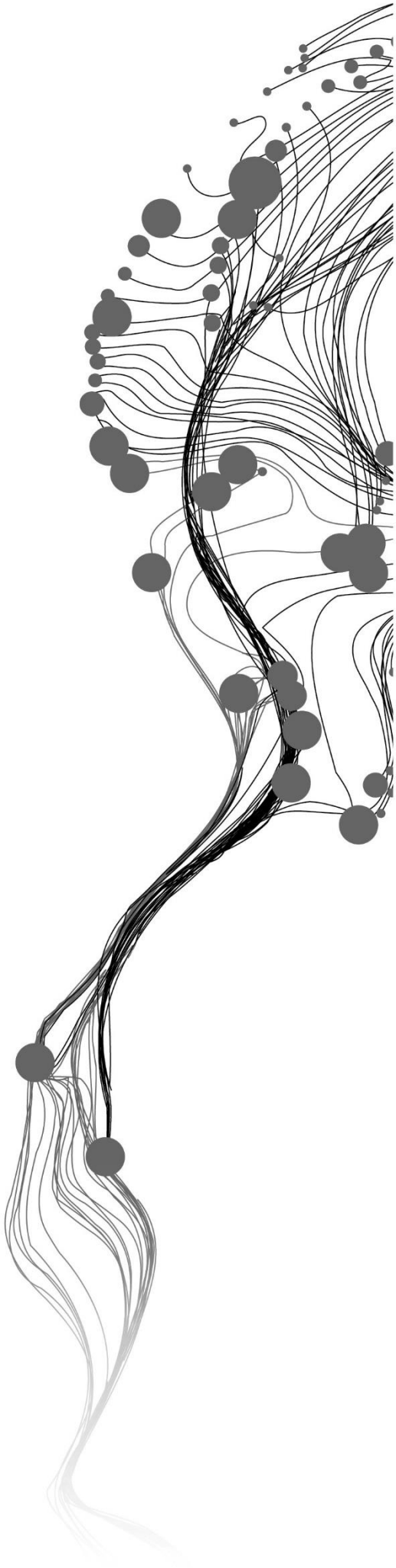
A SOLUTION TO ANALYZE MOBILE EYE-TRACKING DATA FOR USER RESEARCH IN GI SCIENCE

YUHAO JIANG
June, 2020

SUPERVISORS:
dr. C.P.J.M. van Elzakker
dr. P. Raposo
dr. F.O. Ostermann



A SOLUTION TO ANALYZE MOBILE EYE-TRACKING DATA FOR USER RESEARCH IN GI SCIENCE

YUHAO JIANG

Enschede, The Netherlands, June, 2020

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
dr. C.P.J.M. van Elzakker
dr. P. Raposo
dr. F.O. Ostermann

THESIS ASSESSMENT BOARD:
prof.dr. M.J. Kraak (Chair)
dr. P. Raposo (First Supervisor)
dr. F.O. Ostermann (Second Supervisor)
dr. P. Kiefer (External Examiner, Geoinformation Engineering, ETH Zurich)

ABSTRACT

Mobile eye-tracking has enabled GI user studies to be conducted in real-world environments, where the usability of mobile applications presenting spatio-temporal information and the cognitive process during the interaction with the information can be studied in a realistic context. But the dynamics in the real-world environments challenge the analysis of the data, and the standard solutions provided by eye-tracker vendors don't necessarily fit the need of GI user research. This thesis attempts to develop a prototype solution that assists the analysis of mobile eye-tracking data collected with a mixed-methods approach in GI user research.

The development of the prototype solution follows the user-centered-design approach. Requirements are formulated based on a literature review on the application of mobile eye-tracking in GI user studies, the current analysis practice, and existing analytical solutions. The implemented first-stage prototype solution consists of a fixation mapping component, a screen-recording processing component, and a think-aloud data processing component, and provides possibilities for synchronizing the processed data. It attempts to automatically map fixations to real-world objects and screen contents, and (semi-)automatically process the think-aloud data with a transcription-segmentation-encoding pipeline. The results from these components, and location data (GPS measurements during the eye-tracking session), can be synchronized and analyzed together. The prototype solution is demonstrated and preliminarily evaluated with a case study. The case study data was originally collected to evaluate a mobile application aiming to assist geography fieldwork education. In the case study, mobile eye-tracking data, together with screen recording videos, think-aloud audios and GPS recordings are processed and analyzed with the prototype in an exploratory study that aims to describe the interaction between the application and the environment, and to discover usability issues with the application. The analysis explores the distribution and sequence of fixations, identifies usability issues from think-aloud protocols, and describes the test person's fieldwork learning process with synchronized fixation-verbalization-location data.

The prototype solution is able to map fixations and encode think-aloud protocols with reasonable consistency compared with manual processing results. By processing and integrating data collected with a mixed-methods approach, it can assist the exploration of the linking process between the environment, the representation of it, and the mental map as people interact with geographic information in a real-world environment.

ACKNOWLEDGEMENTS

First, I'd like to express my gratitude to dr. Corné van Elzakker. He's been the "super teacher" that I look up to, and it was a pleasure to work on my thesis under his supervision. As I hit a major problem right at the beginning of the proposal phase, he helped me to shape the direction of the thesis project. Along the thesis journey, all the way up to his retirement, with detailed feedbacks and inspiring discussions, he continued to steer me, helped me to become more "concrete", and also encouraged me to develop and play with my own ideas. With his patience and support, I somehow found my feet after a wobbly start, and went on to actually have some fun with the project.

I'm very grateful to dr. Paulo Raposo for his guidance, support, and lots of patience. Especially during the last few months, having regular meetings and discussions with him in the chaotic corona-time meant a lot to me. His feedback and encouragement helped me to finish the writing without getting overly frustrated – I never knew I could write a thesis this long. I would also like to thank dr. Frank Ostermann for willing to step in and take over as one of my supervisors.

Many thanks to Xiaoling Wang, for sharing the GeoFARA data for my case study, and for the discussions we had along the way. It's a small world when two BNU alumni met at ITC and cooperated on each other's projects. Also my appreciation to her husband Simon, who went the extra mile(s) to Enschede and make sure GeoFARA could run smoothly for my case study.

I'd like to thank ITC for a great opportunity and two wonderful years, and the ITC Excellent Scholarship Programme for funding my studies. I had fun with the courses since day one.

My special thanks to Junhui Mao for being a dear friend. We've supported each other through the courses and the thesis. Over the two years, we've shared ideas and dreams, passions and frustrations, so many supermarket trips, and quite a lot of junk food ;-) It's been amazing to have a close friend like her along the way.

My final thanks go to my parents. Over the years, they have given me all the freedom to go for the things I want to pursue, whether it's studying for a degree at the other side of the world, or doing somersaults and walking on my hands. They obviously had concerns of me living solo far away from home, and they could not stress "be careful" enough when I showed off to them my newest stunts. I've always been a stubborn kid and I've grown up to be a very different person than the one they had wished for, but they've always been stimulating and supportive in my passions and decisions. I hope I can make them proud.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and problem statement

Before the developments of wearable eye-trackers, studies have been performed in labs with screen-based fixed eye-trackers where the cognitive aspects of map reading are investigated via visual attention (Krassanakis & Cybulski, 2019). These studies have proven the value of eye-tracking data (often combined with data collected within a mixed-methods approach) in studying interface design or spatial knowledge acquisition even though the experiments could not always be held in realistic contexts of use. The development of wearable mobile eye-trackers has enabled the interaction with geographic information to be studied in the real environment, where the participants solve location-based spatial tasks, possibly interacting with spatio-temporal information presented on a mobile display. Mobile eye-tracking has been applied, for instance, to evaluate the usability of mobile navigation applications and interfaces (Bauer & Ludwig, 2019; De Cock et al., 2019; Ohm, Müller, & Ludwig, 2017), to investigate the influence of landmark and navigation aids on wayfinding behaviors and strategies (Brügger, Richter, & Fabrikant, 2017, 2019; Schnitzler, Giannopoulos, Hölscher, & Barisic, 2016), and to model the process of spatial knowledge acquisition such as self-localization (Kiefer, Giannopoulos, & Raubal, 2014) and route-learning (Wenczel, Hepperle, & von Stülpnagel, 2017). The data has provided insights of visual behaviors and strategies by revealing the allocation of visual attention while participants perform a spatial task. Together with other data collected within a mixed-methods approach, the discoveries in visual attention can be further supported and explained, and a more comprehensive view can be obtained regarding the visual and physical behaviors, as well as the mental process during task execution.

Although manually inspecting the recorded video can also lead to useful observations and insights (e.g. Koletsis et al., 2017), many studies perform qualitative or quantitative analysis on eye-movement metrics derived from the data (for example, the statistical tests in Schnitzler et al., 2016 and the mixed linear model approach in Wenczel et al., 2017). Such analysis needs support from a processing – analysis pipeline that transforms raw gaze data to meaningful metrics. The problem is that the constantly changing environment has brought challenges to the processing and analysis of mobile eye-tracking data. Unlike screen-based eye-tracking where screen-coordinates can be extremely helpful at identifying the map or geographical features being looked at on both static and dynamic/interactive maps (Göbel, Kiefer, & Raubal, 2019; Ooms et al., 2015), there is no such common reference frame in mobile eye-tracking data, making it much more difficult to record *what* is being looked at – which is often the starting point of the succeeding analysis. The standard analysis solutions provided by eye-tracker vendors, aiming for more general purposes, do not necessarily fit the analytical needs in order to answer research questions related to e.g. map use or spatial knowledge acquisition. For example, the metrics calculation and analytics modules of the vendors' software suites are often based on areas of interest (AOIs) as a collection of pixel locations (SensoMotoric Instruments GmbH, 2017; Tobii Pro, 2019b), whereas the focus of GI science research is often on the object level (i.e. the objects present in the environment and their correspondents on the mobile display). Manually registering every fixation into the corresponding pixel location on reference images and delineating AOIs on them can be very laborious and time-consuming (Ohm et al., 2017; Wenczel et al., 2017). While automated fixation mapping tools are offered by some eye-tracker vendors, such as Tobii's Real World Mapping and SMI's Automated Semantic Gaze Mapping (SensoMotoric Instruments GmbH, 2017; Tobii Pro, 2019b), they have been reported to fail to map

fixations when the scene is dynamic (Herlitz, 2018; Utebaliyeva, 2019), which is very common in GI user research, for example when the participant performs a navigation task in the environment. In screen-based eye-tracking studies, other data collected within the mixed-methods approach (e.g. screen-logging, thinking aloud) has directly assisted the analysis of eye-tracking data by associating fixations with map or geographic features (Göbel et al., 2019; Ooms et al., 2015) or supporting discoveries in the eye-tracking data (e.g. Jones & Weber, 2012). Such integration, or synchronized analysis, is not supported by the currently available solutions either. A more automated and integrated analysis process targeting the needs of GI science research will ease some labor off the analysis and possibly extract more information from the data that leads to insights regarding the use and interaction with geographic information in the environment.

The research problem of this thesis can be described as a gap between the existing analytical solutions for mobile eye-tracking data and the required information to answer the research questions related to map use or spatial knowledge acquisition that are being addressed with the help of mobile eye-tracking data. This thesis will focus on the development of a first-stage prototype solution to facilitate the analysis of mobile eye-tracking data for GI science research purposes. The prototype solution aims to add automated elements and attempts to integrate and analyze data within a mixed-methods approach involving mobile eye-tracking data, and a case study will be carried out as a proof-of-concept demonstration and preliminary evaluation of the prototype solution.

1.2. Research objective and questions

The overall objective of this thesis is to develop a first-stage prototype solution to help analyze mobile eye-tracking data collected for GI user research. The overall objective can be achieved by answering the following research sub-questions.

1. What are the requirements for the solution in order to enable it to facilitate analyzing mobile eye-tracking data for GI user studies following a mixed-methods approach?
   - What are the typical research questions being addressed with the help of mobile eye-tracking data in a mixed methods approach and what kind of information is needed to answer those research questions?
   - What is the current state-of-the-art analysis practice and what kind of information can be derived with it? What are the limitations of existing analytical solutions?
   - What additional functionalities are needed for an improved prototype solution in order to facilitate the analysis?
2. How can a prototype solution be designed and implemented in order to address the identified requirements?
3. How can the prototype solution assist the analysis of mobile eye-tracking data to answer the relevant research questions?
   - What information can be extracted with the prototype solution and what is its advantage in extracting the information comparing to the existing analytical solutions?
   - How can the prototype solution be used in the analysis of mobile eye-tracking data to answer the relevant research questions?

1.3. Organization of the thesis

To achieve the research objective and answer the research questions, the methods applied will be based on a User-Centered Design approach (van Elzakker & Wealands, 2007). A prototype solution will be developed based on requirements identified from literature, and it will be demonstrated and preliminarily evaluated with a case study.

The rest of the thesis consists of 6 chapters.

The following chapter is a literature review that discusses the application of mobile eye-tracking in GI science and the current analysis practice regarding mobile eye-tracking data. It also provides an overview of existing analytical solutions. By presenting some typical research questions in the geoscience domain and the existing analytical solutions, it provides a background for the thesis and serves as a starting point for the prototype development. The third chapter presents the adopted methodology framework of this thesis, including a brief introduction to the case study. The fourth chapter describes the design and implementation of the prototype. It starts from formulating the requirements based on the literature review. These identified requirements are transformed into a design of the prototype: the components needed to process eye-tracking data and the possible processing and integration of other data within the mixed-methods approach. And then the implementation details of the prototype are also discussed, including the supporting technologies that the implementation is based upon. The fifth chapter demonstrates the use of the proposed solution and with a case study where mobile eye-tracking data collected in another GI user study project is analyzed with the prototype solution. The chapter presents the information derived with the help of the prototype, and demonstrates how the information can be visualized and analyzed to answer the research questions of the original project. The sixth chapter presents a preliminary (technical) evaluation on the functionalities of the prototype solution where the prototype solution is compared with the current analysis practice and evaluated for its performance. Further analysis possibilities beyond the case study and limitations of the prototype are also discussed in this chapter. The final (seventh) chapter summarizes the thesis work by presenting conclusions, answering the research questions and providing recommendations for further research and solution development.

2. MOBILE EYE-TRACKING IN GI USER RESEARCH: APPLICATION AND ANALYSIS PRACTICE

2.1. Introduction

This chapter is a literature review on the application of mobile eye-tracking in GI user research and on the available solutions to analyze mobile eye-tracking data for such purposes. It will serve as a starting point to identify the needs and requirements for the prototype solution to be developed later in the thesis. This chapter starts with a brief introduction of mobile eye-tracking (Section 2.2). It is followed by a review on the applications of the mobile eye-tracking technique in GI user research, mainly focusing on the research questions they try to answer and the analytical approach they take regarding the mobile eye-tracking data (Section 2.3 and 2.4). A summary of available analytical solutions, both proprietary and open-source, is presented at the end (Section 2.5).

2.2. Mobile Eye-tracking

The eye-mind hypothesis suggests that cognitive processes and strategies can be reflected through visual attention (Just & Carpenter, 1976). Eye trackers can record the movement of the eyes and have been used to study visual attention allocation. There are two types of eye-trackers: screen-based and mobile. As opposed to screen-based eye-trackers where the stimuli are displayed on a screen and the test persons are fixed in front of it (Figure 2-1a), mobile eye-trackers are wearable devices that enable the test persons to move freely in the environment while their visual attention is recorded together with a scene video of what they see (Figure 2-1b). Due to its mobility, mobile eye-trackers have been used in different fields of studies that require in-situ experiments, for example in marketing, sports and human-machine interactions (Wan, Kaszowska, Panetta, A Taylor, & Agaian, 2019).

Figure 2-1 Screen-based and wearable eye-trackers. a) a screen-based eye-tracker fixed at the bottom of the screen; b) a mobile (wearable) eye-tracker (source for both pictures: Tobii Pro, 2015a)

Eye-trackers record the basic eye movements as gazes. Gaze points are recorded as the instantaneous location of regard on the stimulus (Tobii Pro, 2015c). The frequency of gaze points registration depends on the sampling rate of the eye-tracker. To better interpret the eye movements, raw gaze point data are often filtered (classified) into eye movement events such as fixations, saccades, smooth pursuits and blinks. Among these events, fixations are the most commonly used event in mobile eye-tracking studies.

- *Fixation:* A fixation represents a cluster of gazes when the eye stays relatively still on a target. Each fixation has a spatial location on the image plane, a start timestamp, and a duration. Although they are "reconstructed" from gaze points by a mathematical algorithm (i.e., fixation filter) instead of directly measured, they are considered as meaningful episodes of attention. The target being fixated corresponds to the target currently being processed (Just & Carpenter, 1976).

- *Saccade:* A saccade is a rapid eye movement that happens between fixations where the attention is switched to a new target (Fischer & Ramsperger, 1984). Because of the fast movement of the eye during a saccade, information intake and processing mostly don't take place. The sequence of saccade-fixation-saccade is defined as scan-paths, which are often used to measure information search (Goldberg & Kotval, 1999).
- *Smooth pursuit:* A smooth pursuit takes place when the eye follows a moving target. Saccades are often coupled with the pursuits to pick up and follow a moving target (Kowler, 2011).
- *Blink:* A blink is an involuntary closure of the eye. Blinking usually cause 5-10% loss of data (raw gaze points) during a recording session (Tobii Pro, 2019b). Blink rate and latency are also used to indicate mental effort and cognitive load (Zagermann, Pfeil, & Reiterer, 2016).

2.3. Mobile Eye-tracking in User Research in GI Science

As spatio-temporal information is often communicated through mobile displays, user research of such applications and the interactions with them is also conducted in realistic use contexts where people solve spatio-temporal tasks in a real environment. Mobile eye-tracking is used to record the visual attention of people interacting with these products to answer research questions regarding the use and interactions with geographic information and spatial knowledge acquisition in the environment.

There are generally two types of research questions that are addressed with the help of mobile eye-tracking: one focuses more on the design aspects of the applications communicating spatio-temporal information (often on a mobile display), the other focuses more on the cognitive aspect as people interact with these products in a real environment.

The first type of research questions mainly addresses map or application design issues and evaluates the usability of different map or application designs. They focus on the elements being inspected, such as which element receives more visual attention and which kind of map design results in higher cognitive workloads during use. Ohm et al. (2017) used the amount of visual attention to the screen as an indicator of efficiency to evaluate abstract navigation interfaces. Bauer and Ludwig (2019) compared detailed maps with schematic maps in indoor wayfinding by comparing the visual attention spent on the navigation instructions and the time needed for orientation. Apart from maps, written and photo-based navigation instructions and the corresponding mobile applications were also studied and evaluated for usability (De Cock et al., 2019), in which eye movement measures (e.g. mean fixation durations, revisits counts) were used as indicators for mental efforts.

The second type of research questions mainly looks into the cognitive processes and strategies as people solve a spatial task in a real environment with or without a map as aid. They focus on describing, explaining and modelling the process. They investigate the external and human factors that influence the strategies and performance, and how the influence is reflected through visual attention. Apart from what is being looked at, the procedure of such attention allocation, and the cognitive interplay to associate the environment and the display are also at focus. Kiefer et al. (2014) studied the distribution and sequence of visual attention between map symbols and visible landmarks during the self-localization process ("given spatial scenery, identify one's position in a spatial reference frame"), and concluded that more matches between map symbols and corresponding landmarks resulted in more successful task completion, suggesting a more successful self-localization strategy. Wenczel et al. (2017) studied the effect of learning intentions (incidental or intentional) on gaze behaviors during outdoor navigation. Visual attention to landmarks, as indicated by total fixation durations, was compared to indicate different spatial knowledge acquisition strategies under different learning intentions. Schnitzler et al. (2016) compared visual behavior and wayfinding decisions as people navigate with mobile maps, paper maps or no maps. They used the

distribution and frequency of fixations to depict the interplay between the navigator, the navigation device and the environment during an indoor wayfinding experiment, and looked into the characteristics of decision points and navigation devices that led to more attention for navigation aids. Franke and Schweikart (2017) compared navigation performance using maps with and without landmark information to study whether having landmarks on maps results in more attention for the corresponding landmarks in the reality and a more sustainable imprint on the cognitive map. Brügger et al. (2017) studied aided and unaided wayfinding by comparing the egocentric directions of participants' visual attention during the processes. They compared the directional distribution of visual attention to conclude that during unaided wayfinding people looked backwards more in order to re-construct the spatial scene they had travelled during the previous aided navigation phase. A similar aided-unaided navigation experiment setup was later used to study the influence of automation level of navigation system behavior on human navigation behavior where the duration of fixations was used to indicate the cognitive function level along the navigation route (Brügger et al., 2019).

For both types of research questions, the underlying cognitive process can be described as a mental process that links the reality (i.e., environment), the representation of it (e.g. a map on a mobile phone) and the cognitive map of the person (Delikostidis, 2011). This process is largely supported by visual attention, such as looking for clues in the environment. The attention allocation process between the environment and the representation can be reflected directly by eye movements. While the interaction with the cognitive map cannot be directly measured by eye-tracking data, it can be inferred from other data such as think-aloud recordings or mental map drawing.

Indeed, mobile eye-tracking is often applied within a mixed-methods approach to be able to better answer such research questions. Thinking aloud can be used alongside mobile eye-tracking to help discover intentions and strategies of the test persons. Both concurrent and retrospective thinking aloud have been applied to compare navigation strategies between groups (Koletsis et al., 2017; C. Wang, Chen, Zheng, & Liao, 2019). Verbal protocols of think-aloud sessions also provide information to explain both visual and physical behaviors such as why a participant missed a target or got lost (Koletsis et al., 2017). As the studies are often conducted outdoor and involve locomotion, location data (GPS recordings) can also be collected to integrate locomotion and spatial context into the analysis. Kiefer, Straub, and Raubal (2011, 2012) demonstrated an analysis of location-based mobile eye-tracking, where GPS data helped to reveal map reading behaviors. They mapped locations where a map was most needed, and explored the locomotion speed during map reading as an indicator of map use strategy. Unlike screen-based studies where the screen stimuli are automatically recorded and user interactions, such as mouse and keyboard events, can be logged and integrated into the analysis (e.g. Ooms et al., 2015), so far, screen-recording of the mobile display and user interaction logging are relatively rare in existing mobile eye-tracking studies, even though the content on the mobile display is often of interest. Some studies incorporated user-logging elements in their test applications and participants were asked to click a button once they understood the navigation instruction or successfully oriented themselves (Bauer & Ludwig, 2019; Ohm et al., 2017). This kind of user-logging has provided important information regarding the completion time of sub-tasks. Other user research methods, such as interviews (Franke & Schweikart, 2017), questionnaires (Bauer & Ludwig, 2019; De Cock et al., 2019) and memory recall tests (Franke & Schweikart, 2017) are also performed and their results can be referred to the results from mobile eye-tracking to support and complement each other and to discover relationships.

2.4. Analytical practice to for mobile eye-tracking data in GI user studies

A typical analytical pipeline for raw gaze data usually starts with de-noising and filtering gazes into fixations (also known as eye-movement event detection or classification), followed by the collation of fixation-related information and then a visual or statistical analysis of the fixation data (Kiefer, Giannopoulos, Raubal, & Duchowski, 2017). Gazes are filtered into fixations based on whether the eye stays relatively still. For example, the I-DT dispersion threshold filter detects a fixation when the consecutive gaze points are distributed within the dispersion threshold; the I-VT velocity threshold filter detects a fixation when the (angular) velocity of eye movement is under a given threshold (Salvucci & Goldberg, 2000). This computation is based on the coordinate system of the eye-tracker (i.e., the movement of the eye is calculated independent of the movement of the target being looked at). In many fields of studies where mobile eye-tracking is applied, gaze filtering is not of primary interest to the researchers, as they are more interested in questions such as *what is being looked at*, instead of how the eyes move in respect to the head of the participant (Niehorster, Hessels, & Benjamins, 2020); they often work on "fixations detected by the software" (for example, in Franke & Schweikart 2016; Wenczel et al. 2017).

Similar to the analysis of screen-based eye-tracking data on interactive map use, the analysis of mobile eye-tracking data can also be classified into the two major categories: content-independent and content-dependent analysis (Göbel et al., 2019). In the case of mobile eye-tracking, the difference between these two types of analysis lies in whether fixations are mapped to the objects (both in the environments and on the mobile display) being looked at.

The first type of analysis is performed with aggregated metrics without distinguishing the targets of fixations. For example, in the study of Brügger et al. (2019), the data was segmented to sections based on sections along the task route and fixation metrics were aggregated per section without considering what the visual attention was allocated to. In their study, the descriptive summary statistics showed that mean fixation durations could be used to identify different behaviors and cognitive function levels along the route.

On the other hand, in many studies the object being inspected is at focus, especially when maps are involved, as it is often of core interest to the study to know *what* is being looked at as people associate objects in the environments with the representations of them on the visual displays. In these cases in which the analysis is content-dependent, each fixation is associated with a target object before metrics are calculated. Because objects continue to change positions in the scene video and might move out of view, fixations are often mapped to one or more static reference images (also known as "snapshots") where all objects of interests are present. Although the software suite from eye-tracker vendors provide some degrees of automation in doing this, the mapping process is still mostly manual and laborious (Kiefer et al., 2017).

One approach to map fixations is to use a scene image (i.e., a frame from the video recording) as reference and register each fixation to its corresponding location on the reference. Areas of Interest (AOIs) can then be defined on the reference image and AOI-based metrics can be calculated for succeeding analysis. Visualizations such as heatmaps and gaze plots can also be created on the reference image (see Section 2.5 and Figure 2-4). Yet due to the dynamics of the recordings, often too many scene images are needed to address all the objects that appeared in the video. Identifying AOIs manually on the excessive amount of reference scene images adds to the labor of fixation mapping (Ohm et al., 2017). At the same time, pixel-level precision is not always needed when the focus is on the *objects* being looked at. Another approach is to use a schematic image as reference, in which abstract representations, such as "placeholder" boxes, represent different objects or object categories, both in the environment and on the mobile display. In

order to study the allocation of visual attention as people associate real-world objects with their representations, the mobile display itself often stands out as an object of particular interest. The display can be treated as a whole (Schnitzler et al., 2016), or divided into different sections (e.g. map section and navigation instruction section; Bauer & Ludwig, 2019; Ohm et al., 2017). Sometimes, fixations are also mapped to the exact corresponding locations or map symbols on the (paper) map for a more detailed analysis when the accuracy of the eye-tracker allows it (Franke & Schweikart, 2017; Kiefer et al., 2014). On the environment side, objects of interests are usually potential landmarks, including but not limited to buildings and signages (De Cock et al., 2019; Franke & Schweikart, 2017; Schnitzler et al., 2016; Viaene, Vansteenkiste, Lenoir, De Wulf, & De Maeyer, 2016). Nonetheless, both approaches of fixation mapping are laborious and time-consuming due to the huge amount of fixations to be mapped, which has also become a constraint for the number of participants recruited and, in turn, this is restricting the application and credibility of statistical methods (Bauer & Ludwig, 2019).

Although, e.g. for discovering usability issues and formulating research hypotheses, the analysis of mobile eye-tracking data can be qualitative through inspecting the videos and annotating high level behaviors (e.g. looking at map, confirming landmark, as in the study of Koletsis et al., 2017), the eye-movement data is often analyzed with metrics and statistics. Fixation-related metrics are often used as measures for the visual interpretation process. Fixations can be analyzed by their distribution and sequence (Kiefer et al., 2014). Distribution related metrics, such as fixation counts, frequency, total and mean fixation durations, can suggest what parts of the map or the environment are attended more (Bauer & Ludwig, 2019; Kiefer et al., 2014; Ohm et al., 2017; Schnitzler et al., 2016; Wenczel et al., 2017) and may be indicative of the cognitive function level when processing the information (Brügger et al., 2019; De Cock et al., 2019). The sequence of fixations describes the process of obtaining and processing such information and is often depicted by metrics such as (map) revisit counts (De Cock et al., 2019), and number of matches between object in the environment and the corresponding map symbols (Kiefer et al., 2014). Although saccade-related metrics are often used in screen-based eye-tracking studies as a measure of visual search (Liao, Dong, Peng, & Liu, 2017), they are less common in the analysis of mobile eye-tracking data due to the difficulty to distinguish saccades and smooth pursuits when both the head and the stimuli are moving in a dynamic setup (De Cock et al., 2019; Schnitzler et al., 2016). Because the detection of eye-movement events is based on the movement of the eye in the coordinate system of the eye-tracker, it can be prone to error when the object and/or the head is moving. Especially, when the eye follows a moving object, smooth pursuits are often classified as saccades or fixations (Olsen, 2012; Tobii Pro, 2019b).

While visual analysis approaches such as heat maps and scan-path visualizations are common for screen-based studies apart from statistical analysis (Blascheck et al., 2017), they are less commonly used to analyze mobile eye-tracking data because these visualizations often require high-precision fixation mapping where fixations are registered into the exact corresponding points on scene images.

The processing and analysis of other data collected within the mixed-methods approach are often carried out independent of the analysis of eye-tracking data until their results are referred to each other. For example, think-aloud protocols are processed (mostly manually) with the transcription-segmentation-encoding workflow, and the coding results can be analyzed with code frequency (Koletsis et al., 2017; C. Wang et al., 2019); the transcripts are also directly used to discover and support findings in the eye-tracking data in exploratory analysis (Koletsis et al., 2017; Utebaliyeva, 2019).

2.5. Available analytical solutions

Eye-tracker vendors provide software suites that process and analyze the collected data, and they are often the choice of analysis of researchers. SMI and Tobii are two popular solutions used by researchers from varying backgrounds (Wan et al., 2019). Table 2-1 lists the main functionalities provided by Tobii Pro Lab and SMI BeGaze.

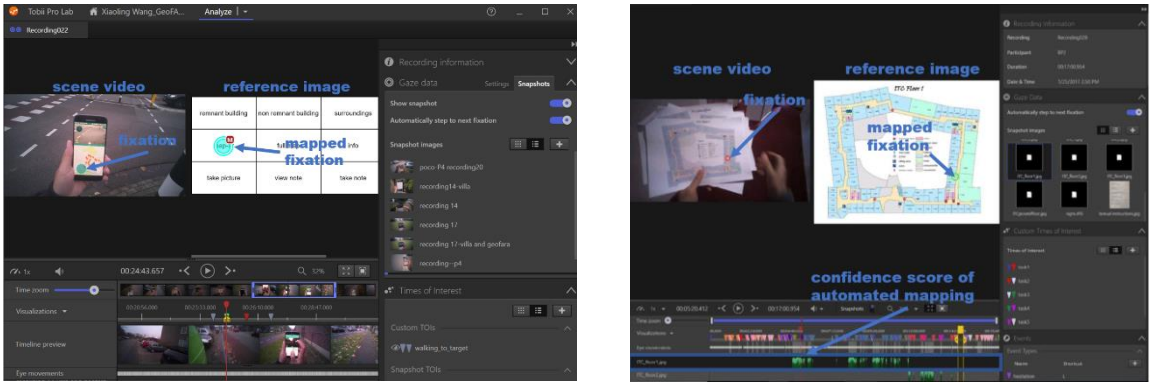
Table 2-1 Vendors' analytical solutions

Functionality Group	Tobii Pro Lab (v1.123)	SMI BeGaze (v3.7)
Recording replay	gaze (fixation) video overlay on stimulus	gaze (fixation) video overlay on stimulus
Gaze and fixation mapping	manual mapping and automated mapping of raw gazes and filtered fixations to snapshots	manual mapping and automated mapping of raw gazes and filtered fixations to snapshots
AOI definition	static: static shape defined on image stimuli or snapshots; dynamic: shape defined on video keyframes and interpolated on frames in between	gridded: static content-independent grids overlaid on stimuli/snapshot; static: static shape defined on image stimuli or snapshots; dynamic: shape defined on video keyframes and interpolated on frames in between
AOI-independent metrics	saccade metrics (saccade count, peak velocity of saccade, saccade amplitude, time to first saccade)	fixation metrics (fixation count, fixation frequency, fixation duration); saccade metrics (saccade count, saccade duration, saccade amplitude, saccade velocity, scan path length); blink metrics (blink count, blink frequency, blink duration)
AOI-independent visualization	heatmap, gaze plot.	heatmap, focus map, bee swarm, scan path
AOI-based metrics	fixation metrics (fixation duration, fixation count, time to first fixation, duration of first fixation); visit and glance metrics (visit duration, visit count, glance duration, glance count); saccade metrics (saccade count in AOI, time to entry/exit saccade, peak velocity of entry/exit saccade)	fixation metrics (fixation duration, fixation count, first fixation duration, time to first fixation); visit and glance metrics (visit time, revisit count, glance duration, glance count, diversion duration); saccade metrics (time to first saccade); sequence metrics (AOI visit sequence, AOI transition matrix)
AOI-based metrics visualization	not natively supported	AOI sequence chart, binning chart, proportion of looks chart
Think-aloud recording and processing	not natively supported	includes recording module for retrospective think-aloud, no analysis functionalities.

In recording replay, fixations (gazes) are overlaid on the scene camera video. Researchers can add events to the replay timeline, which allows them to add annotation about e.g. higher-level behaviors, comments, codes for think-aloud protocols, or time of interest (TOI). It can facilitate the visual interpretation of the recordings. An example of the recording replay and timeline annotation is shown in Figure 2-2. In the replay view, the fixation is represented by a cyan circle. In the timeline, there is one TOI "walking_to_target", and a customized event "keyword".

Figure 2-2 Example of recording replay in Tobii Pro Lab, including fixation overlay, TOI, timeline and customized event

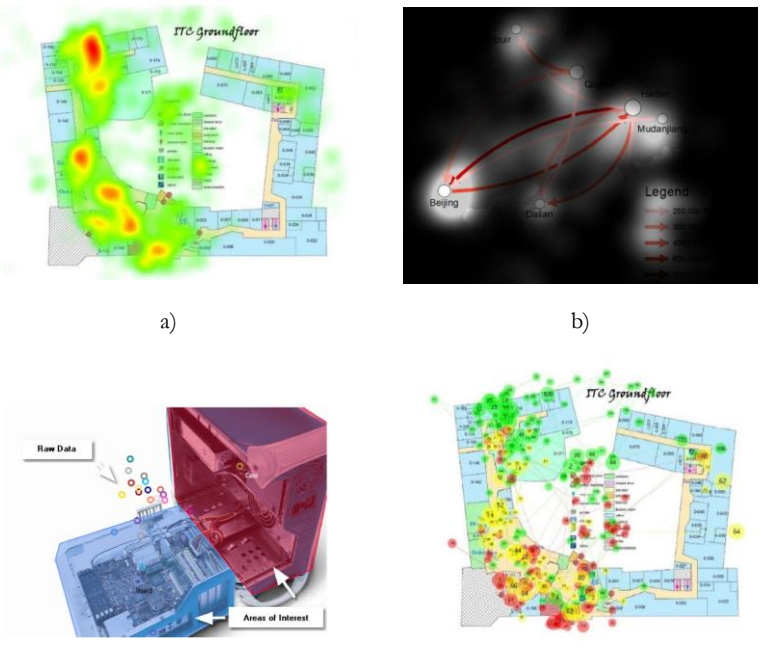
Gaze and fixation mapping is an important functionality that allows researchers to associate gazes and fixations with stimuli in the real world. Gazes and fixations are registered to a reference image (a snapshot). Manual mapping is normally performed on fixations due to the relatively smaller amount of fixations to be mapped, with respect to gaze points. During manual mapping, each fixation in the recording video is manually mapped to a reference image. The reference image can either a scene image or a schematic image. An example of manual mapping to a schematic reference image in Tobii Pro Lab is shown in Figure 2-3a. Automatic mapping, as the example of Tobii's Real World Mapping (RWM) tool, utilizes image-matching algorithms that find the corresponding part of the reference image in the frames of eye-tracking videos (Herlitz, 2018). It is performed on gaze points, and fixations can be calculated on the snapshot based on the mapped gazes. It is estimated that automated mapping is approximately 5 to 10 times faster compared to manual mapping for fixations (Tobii Pro, 2015b). To maximize its performance, flat (i.e., without perspective) and high-resolution reference images are preferred (Tobii Pro, 2019a). The mapping workload can be reduced significantly with automated mapping when the target in the video is relatively big, planetary and stable, for example, a paper map (Li, 2017; C. Wang et al., 2019). However, the accuracy of such automated mapping tools might be far from ideal in some cases with more head movements and perspectives in the recordings (Herlitz, 2018) and Tobii's RWM has been reported to be not useful when the environment is highly dynamic (e.g. when the participant is walking; Herlitz, 2018) and the target is relatively small (e.g. a smartwatch; Utebaliyeva, 2019). An example of the automated mapping in Tobii Pro Lab is shown in Figure 2-3b.



a) b)

Figure 2-3 Manual and automated fixation mapping in Tobii Pro Lab. a) manual mapping to a schematic reference image; b) automated mapping to a paper map (adapted from Li, 2017)

AOI-independent visualizations are provided in two categories: density visualizations (heatmap, focus map) and scatter visualizations (gaze plot, scan path, bee swarm). For mobile eye-tracking data, these visualizations are only available after gazes and fixations have been mapped to a reference image. Heatmaps and focus maps are kernel density estimations of gazes or fixations that show the distribution of visual attention by changing the color or transparency of the background image based on the amount of attention received (e.g. measured by fixation count or fixation duration). Gaze plots, scan path, and bee swarm visualize individual gazes or fixations. Bee swarm plots show raw gazes as colored circles or other cursor shapes. Gaze plots and scan path graphs show the sequence of visual attention in which the fixation sequence is represented as numbered point symbols connected by saccade lines, where the sizes of the point symbols may be made proportional to the duration of the fixation, and the color of the point symbols can represent different participants. Examples of the AOI-independent visualizations are shown in Figure 2-4.

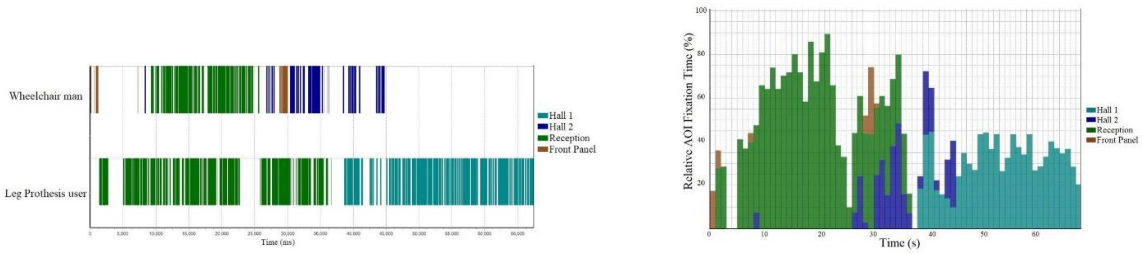


a) b)

c) d)

Figure 2-4 Examples of AOI-independent visualizations. a) heatmap of mapped fixations on a floor plan (Source: Li, 2017); b) focus map of visual attention on a flow map (Source: Dong, Wang, Chen, & Meng, 2018); c) bee swarm on an image (Source: SensoMotoric Instruments GmbH, 2017); d) gaze plot on a floor plan (Source: Li, 2017).

For mobile eye-tracking data, it is possible to define AOI on the scene camera video or the reference image. Static AOIs defined on the video will remain static, independent of the change of video content, which makes it less useful in a dynamic setting. Dynamic AOIs are defined on the video keyframes and the shapes are interpolated for video frames in between, also independent of the video content and rarely used. Defining AOIs on the reference image enables metrics calculation on mapped fixations. Once AOIs are defined, metrics measuring dwell and transitions between the areas can be calculated and exported. AOI-dependent metrics can be visualized to show the distribution and sequence of visual attention among the defined AOIs. Figure 2-5 shows examples of standard visualizations provided by SMI BeGaze.



a) b)

Figure 2-5 Examples of visualization of AOI-dependent metrics produced with SMI BeGaze. a) AOI-sequence graph that shows the visual attention sequence of two participants in four AOIs along a timeline; b) binning chart that shows relative AOI fixation time on four AOIs along a timeline (Source of both charts: Merino, Riascos, Costa, Elali, & Merino, 2018).

The software suites from eye-tracker vendors provide little support for other data collected with the mixed-methods approach. Although the hardware (e.g. Tobii Pro Glasses 2 and SMI ETG) allows the recording of audio data simultaneously with the video, the analysis of such data is not natively supported. The lack of an automated approach to integrate think-aloud data is also reported as a problem in the analysis of mobile eye-tracking data in general, not limited to GI user research (Wan et al., 2019). Screen-recording of the display during a mobile eye-tracking session is not supported in the vendors' suites either.

The open-source and research communities have also produced analytical solutions for mobile eye-tracking data. They can be vendor-independent and can process data from different eye-tracker models. Apart from reproducing the processing tools of the vendors' software suite (e.g. gaze denoising and eye movement detection), some solutions focus on specific aspects of the analysis such as fixation mapping or automated AOI generation. An inventory of these solutions is shown in Table 2-2.

Table 2-2 Other available solutions

Solution	Presented as	Supported eye-tracker specification	Main functionalities	Additional information
TobiiGlassesPySuite (processing module) (De Tommaso & Wykowska, 2019)	Python library	Tobii	parsing and extracting gaze data; recording management and gaze filtering	The full solution has a controller module for the controlling of Tobii Pro Glasses 2, to form a collection-processing pipeline.

UXI.GazeToolKit (Konopka, 2019)	C#/.NET library with console application	no specification	gaze filtering and data validation	It does not directly depend on SDK from eye-tracker vendors.
GlassesViewer (Niehorster et al., 2020)	MATLAB program with graphic user interface	Tobii	parsing, extracting ad viewing gaze data; gaze filtering	Multiple data streams (pupil size, gyroscope, accelerometer, etc.) can be viewed.
GazeCode (Benjamins, Hessels, & Hooge, 2018)	MATLAB program with graphic user interface	SMI, Positive Science, Tobii, and Pupil Labs	manual fixation mapping	The interface is optimized so that manual mapping fixation to object categories is reported to be approx. two times faster with than with Tobii Pro Lab
Mobile Gaze Mapping (Macinnes, Iqbal, Pearson, & Johnson, 2018)	Python command-line tool	no specification	automated fixation mapping based on feature-matching	Gazes are mapped to corresponding locations on a target stimulus (i.e., an object on a reference image). The reference image needs to be cropped to only include the target stimulus to ensure the performance.
Visual Analytics Tool (Kurzhals, Hlawatsch, Seeger, & Weiskopf, 2017)	program with graphic user interface (closed source)	no specification	dynamic AOI generation with image clustering and interactive labelling	This approach mainly focuses on the analysis of hypothesis-driven experiments in which AOIs can be pre-defined (e.g. poster-viewing)
Computational Gaze-Object Mapping (cGOM) (Wolf, Hess, Bachmann, Lohmeyer, & Meboldt, 2018)	Python command-line tool	no specification	automated fixation mapping to object AOIs with image instance segmentation (Mask-RCNN; He, Gkioxari, Dollár, & Girshick, 2017)	The model was trained with only 72 annotated images, and over 4000 fixations were mapped to object AOIs with approx. 80% accuracy. It demonstrated the potential of object-based semantic fixation mapping using a neural network with only a relatively small set of training data.

2.6. Summary

This chapter provided the background of the thesis. It reviewed the application of the mobile eye-tracking technique in GI user research and the typical research questions addressed with it, and summarized the current analytical practices and available solutions. The typical research questions are mainly related to the usability and design aspects of mobile application, and the cognitive process of spatial knowledge acquisition. The current analytical practices for the mobile eye-tracking data is often based on fixation metrics after manually mapping the fixations to real-world objects and screen-contents of the mobile display. Other data collected with the mixed-methods approach is often analyzed independently of the analysis of mobile eye-tracking data. The currently available analytical solutions are not all useful when it comes to answering the research questions related to interface design and the cognitive process in GI user research: automatic mapping of fixations to real-world objects is not well supported, the screen content on the mobile display cannot be automatedly incorporated into the analysis, and there is no automated processing and integration approach for other data collected with the mixed-methods approach, in particular, think-aloud audio data, to be analyzed together with eye-tracking data. This thesis research will build upon these existing analytical solutions and aims to address the gap by assisting fixation mapping processes with automation that associates fixations to real-world objects and screen display contents, and to integrate data from the mixed-methods approach to the analysis. The needs and requirements identified from existing research will be the starting point of the development of the prototype solution. The next chapter will introduce the general methodology adopted in the thesis regarding the development of the prototype solution.

3. METHODOLOGY OUTLINE

3.1. Introduction

This chapter outlines the research methodology of the thesis. The thesis adopts the User-Centered Design (UCD) approach (van Elzakker & Wealands, 2007) for the development of the prototype solution. A case study is applied to demonstrate the functionalities as a proof-of-concept and to preliminarily evaluate the prototype. Section 3.2 outlines the adopted UCD approach. The background of the case study is introduced in Section 3.3.

3.2. User-centered design and application development

The UCD framework has become one of the guiding principles for designing usable technologies and is often employed in the design of various geoinformation products (Haklay, 2010; Roth et al., 2017). The framework guides the development of applications, taking into account how the application/product can directly support the work of the users. Haklay (2010) presented the UCD cycle for geospatial technologies (Figure 3-1). The project starts with the planning: gathering information on what is needed to ensure the usability of the end product, usability of existing applications, and ideas for new product development. The design and development of the application is an iterative process. It starts with the analysis of user requirements, including the tasks, contexts of use and characteristics of the users, followed by a first-stage prototype of the design, and an evaluation of whether the design satisfies the requirements defined in the first stage. Iteration takes place when the requirements are not fully met: user requirements are then refined, and prototyping and evaluation will follow. The product ready for deployment is the outcome of the iteration process after the requirements are satisfied.

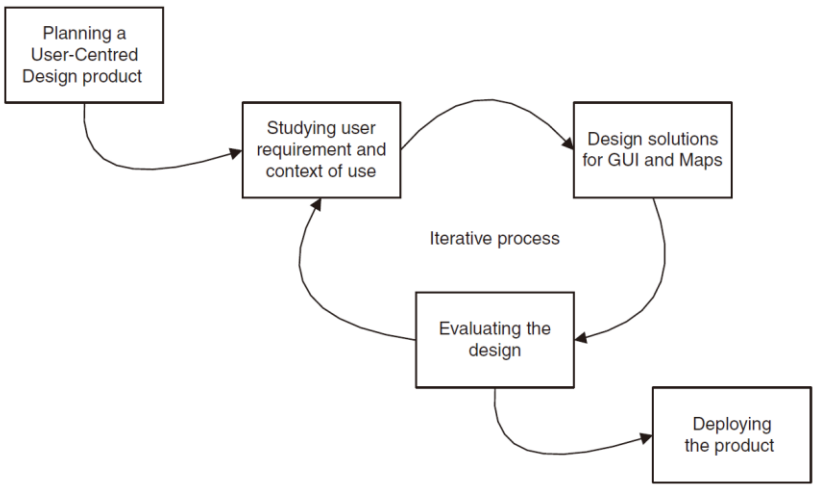


Figure 3-1 A UCD cycle for geospatial technologies (Haklay, 2010, p100)

Usability engineering translates the usability concepts into actions and criteria for developers. For example, the criteria for usable computer programs include effectiveness, efficiency, error-tolerance, learnable and satisfaction (Haklay, 2010). These criteria, often translated to more specific measurements, guide the development process. The main stages of the application development process are in line with the main

stages of the UCD cycle: gathering requirements and needs, development of the application, evaluation with typical/potential users, and finally deployment when the needs of the users have been addressed. Many methods and techniques have been developed for all three stages of the process (see Haklay, 2010, and Delikostidis, 2011 for reviews and summaries on methods and techniques)

The first stage is analyzing and developing requirements. The functionalities of the application should be derived from the needs of the user. An understanding of the potential users is needed in order to develop functionalities for them. The techniques of collecting user needs includes questionnaires and interviews with potential users, and analysis of existing data and statistics, especially when the goal is to improve an existing application (rather than developing a new one; Haklay, 2010). An understanding of the use context is also needed, which can be acquired from qualitative methods such as direct observation. An understanding of the tasks undertaken by the users is often needed to design functionalities to support those tasks.

The prototype solution aims to allow researchers in the GI domain who use mobile eye-tracking to answer their research questions about the use of geospatial technologies in the environment. Because the proposed solution is built upon existing analytical solutions, an analysis of existing literature is the main source of the requirements. The analysis is conducted with a literature review (Chapter 2). A review of the use of mobile eye-tracking in GI user research identifies the typical research questions (the goal: what they use it for), the desired information to solve these questions and the current analysis practice to derive the information from the data (the tasks: what they do with it), and the difficulties of deriving the desired information with available solutions. Requirements for the prototype are formulated based on these findings.

The second stage is the development of the application. During this stage, usability guidelines and design principles in literature can be a reference for the development (e.g. style guide for Tensorflow development; TensorFlow, 2015). A special consideration in this stage is about which parts of the application can be open for customization, and which parts should be encapsulated and hidden from the users, to minimize potential errors during use (Haklay, 2010). During the development stage, limited evaluation and testing can help determine the key elements (e.g. data model, workflow) of the application, especially when the user interfaces have a strong link with the core functionalities of the application.

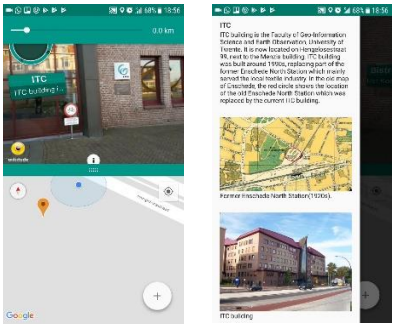
The first-stage prototype to be developed in the thesis will not include (graphic) user interfaces. Its main focus is the processing of the data (mobile eye-tracking, with other data collected with a mixed-methods approach) and providing the possibilities to derive (additional) information more efficiently compared to the existing analytical solutions. The prototype solution addresses the problem from two aspects: the processing and analysis of mobile eye-tracking data, and the processing and integration (analysis) of other data collected within the mixed-methods approach. It aims to encapsulate the functionalities, but will also provide open source code so that everything can potentially be customized for expert users.

The evaluation stage tests if the developed application meets the requirements stated in the earlier stage. A wide range of frameworks, methods, data analysis and collection techniques are available for this stage (MacDonald & Atwood, 2013). Case study is one of the major empirical frameworks for evaluation, where a single use case is intensively examined to yield results that can be generalized to more similar units. The explorative nature of case studies makes them suitable for research applications, and they are often performed when the depth of the examination is preferred over the broadness (Gerring, 2004). The depth is preferred for example to understand the use of existing systems (Haklay, 2010).

For this thesis, the case study is the main method for the evaluation. It includes a proof-of-concept demonstration of the use of the prototype and a preliminary technical evaluation that compares the prototype with the current analytical solutions. The major focus is functionality and reliability (i.e., if the prototype solution can help to answer the research question of the case study; and how it compares to the state-of-art analysis methods). Actual user testing is not conducted because of the preliminary status of the prototype and it does not (yet) have a graphic interface.

3.3. The GeoFARA case study

The case study in this thesis is the eye-tracking session of the evaluation study of mobile application GeoFARA (X. Wang, van Elzakker, Kraak, & Köbben, 2017). GeoFARA ("Geography Fieldwork Augmented Reality Application") is a mobile application designed to support fieldwork in human geography education by combining visualizations (mobile maps) and mobile augmented reality (AR). As a "context-aware" learning tool, its main goal is to assist students to improve their geographical understanding of an urban area. Points of interest (POIs) related to the fieldwork (e.g. buildings) are overlaid through AR, and also marked on an interactive map, so that the user can have both the POI overview on the map and the POI live view through the AR (as floating labels). The information (e.g. text and images) attached to the POIs can be displayed on demand. The AR and map are displayed on a split-screen (Figure 3-2a), which allows the user to perceive the information of the surroundings with the AR and the map at the same time. When the user clicks the POIs on the AR or the map, detailed information of the POIs is displayed (Figure 3-2b), it can contain text, photos, old maps etc.. The user can also take notes or photos, and view the saved notes and photos. The details of the design of GeoFARA can be found in X. Wang, van Elzakker, & Kraak (2017)



a) b)

Figure 3-2 Two main interfaces in the operation prototype of GeoFARA. a) augmented reality and map on a split-screen, POI "ITC" is shown as the green label in the AR view and the orange marker on the map view; b) detailed information on the POI (source: Wang, 2018)

The operational prototype of GeoFARA was evaluated with fieldwork sessions representing the scenario of human geography fieldwork in higher education. The evaluation study was conducted in 2017. The full evaluation session was a combination of pre-fieldwork spatial ability survey and mental map drawing of the fieldwork area, a fieldwork session with mobile eye-tracking, and think-aloud, and post-fieldwork interview and mental map drawing. The goal of the evaluation session was to find out the utility and the usability of GeoFARA in assisting the student to meet the fieldwork objectives (i.e. improving the geographical understanding of an urban area). (The detailed procedures of the evaluation session can be found in X. Wang, 2020).

The goal for the mobile eye-tracking part was to investigate the fieldwork learning process assisted by GeoFARA (i.e., the simultaneous interaction with the environment and GeoFARA) and to discover its usability issues. The eye-tracking session was conducted with Tobii Pro Glasses 2. Audio data were

recorded simultaneously by the eye-tracker. GeoFARA was run on an Android phone. The screen content of the phone was not recorded. The evaluation study was conducted with 3 pilot test persons and 14 formal test persons. The first pilot study was conducted during the development phase of GeoFARA, and the other two pilot studies after the development had been completed. The pilot studies had the same procedure as the formal test.

The fieldwork area was the Schuttersveld area in Enschede, The Netherlands. The area has a history of usage by the textile industry. Although the textile industry has largely collapsed in Enschede, some visible remnants of the industry are still present (e.g. new buildings built on the site of the old textile factories, villa of the factory owner). These remnants are included in GeoFARA as POIs.

A map showing the fieldwork area with the POIs is shown in Figure 3-3. The task for the test persons during the fieldwork was open-ended: to discover the remnants and visible influence of the formal textile industry in the Schuttersveld area. Test persons were expected to discover the remnants of the formal textile industry, compare them with the current geography, and look for visible clues of the influence of the textile industry on the current spatial layout of the area. There were no fixed routes, and test persons could explore the entire area in their own order and at their own pace. During the fieldwork session, test persons were encouraged to speak their thoughts aloud. Because test persons were informed that they would draw their mental maps of the Schuttersveld area before and after the fieldwork session, active engagement of the mental map could be expected during the fieldwork (instead of passively following instructions presented on the app, they were expected to actively explore both the area and the various information offered by the app).

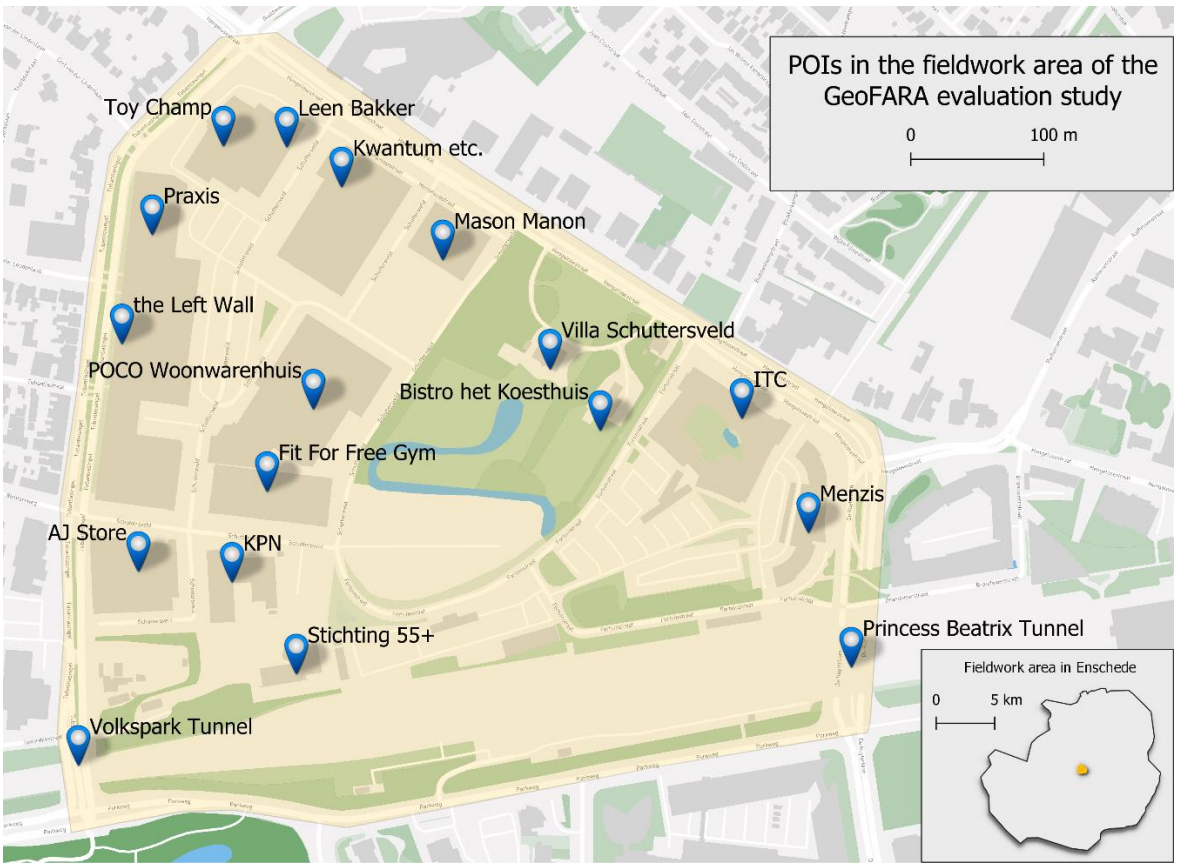


Figure 3-3 Fieldwork area of the GeoFARA evaluation study, Schuttersveld, Enschede

The GeoFARA evaluation study is chosen as the case study because it resembles the typical mobile eye-tracking studies in GI user research: test persons moved in the environment while interacting with a mobile app providing spatio-temporal information and trying to relate the current reality with the (current and historical) representation with active engagement of their own mental maps; regarding data collection, mobile eye-tracking was carried out in a mixed-methods approach together with concurrent think-aloud. The research question of the GeoFARA evaluation also fits into the typical research questions identified in the previous chapter (i.e., evaluate user interface, investigate the cognitive process).

The evaluation sessions had been completed before the start of this thesis project but the research questions (regarding the mobile eye-tracking part) had not been answered yet. The data reflected an authentic status of data collection from a real researcher in GI user research.

3.4. Summary

This chapter presented an overview of the research methodology adopted in the thesis. The design and development of the prototype solution are based on the UCD approach. The requirements source from an analysis of existing literature. A first-stage prototype will be developed based on the requirements on functionalities (processing needs) and will not include a graphic interface. The functionalities of the implemented prototype will be primarily evaluated with a case study. In the case study, mobile eye-tracking data collected from the evaluation session of GeoFARA will be processed and analyzed with the prototype solution, and the prototype solution will be evaluated on functionality and reliability. The case study aims to make generalizations on the potential use of the prototype solution in a wider context of GI user research. The next chapter will present the requirements, design and implementation of the prototype solution.

4. A PROTOTYPE SOLUTION

4.1. Introduction

This chapter presents the design and implementation of the prototype solution. Section 4.2 formulates the requirements for the prototype solution based on the literature review in Chapter 2 and presents the design of the prototype. The implementation framework and the implementation details (including the supporting technologies) are discussed in Section 4.3.

4.2. Requirements

Based on the literature discussed in Chapter 2, the typical research questions being addressed with the help of mobile eye-tracking (in a mixed-methods approach) are mainly related to the usability and design aspects of mobile application, and the cognitive process of spatial knowledge acquisition. Mobile eye-tracking are used in these studies to: a) describe the use and discover usability issues of a mobile application presenting spatial-temporal information; and b) reveal the user's cognitive process as they make interactions between the environment, the representation of the environment and their mental map while performing a spatial task. The literature review has also identified the information needed to achieve these purposes as well as the gap in the analytical solutions. A list of the desired information to be collected, the problems with performing analysis with the current solutions, and the corresponding aspects that need to be considered in the prototype solution is presented in Table 4-1.

Information on visual attention (real-world objects and screen content), mental process and geographical context is needed. The analysis of visual attention is often based on real-world objects and their semantics. Among the objects, the mobile display is an object of particular interest as it presents the geographical information. Apart from whether the attention is on the mobile display, researchers also want to know what on the mobile display is being looked at. To answer the typical research questions, information is needed not only on *what* is being looked at, but also *why* these things are looked at and what is the test person thinking about when making the interactions. Although visual attention can help infer mental processes, the test person's own verbalization provides direct insights from his/her own perspective. It helps to support and explain the visual and physical behavior, it can also reveal issues that not directly visible from visual attention alone, such as difficulties experienced by the test person. And because the experiments are conducted in the real-world environment, the geographical context and locomotion are often heavily involved. Existing analytical solutions are not supporting these needs effectively, especially regarding the fixation mapping and the processing and integration of data from the mixed-methods approach.

Table 4-1 From desired information to prototype design

Desired information	Explanation	Problems with existing solutions	Aspects to consider in prototype
Visual attention on real-world objects	The analysis of the distribution and the sequence of visual attention is based on real-world objects; among the them, the mobile device is an object of particular interest (What *object* is being looked at, for how long, and in which order?)	Mapping fixations to real-world objects is manual and laborious, and current automated mapping solutions cannot handle dynamic scenes well.	A less laborious approach to attach semantic information to gazes or fixations that associates them with objects in reality.
Visual attention on the content of the mobile display	The content on mobile display is often of particular interest because of the spatio-temporal information presented on it. (When the person is looking at the mobile display, what content is being looked at?)	The screen content is often dynamic due to user interactions, and there is no available tool that supports synchronized and automated analysis of screen content and mobile eye-tracking data.	Recording of the screen content and interactions on the mobile display (e.g. as video or action logs), processing of the recorded data and associating the result with the fixations.
Mental process from the test person's perspective	It includes the intention of (visual and physical) behaviors, comments and explanations. (Why does this person look at these things in such a manner? What is this person trying to do?)	The current analysis of verbal protocols is largely manual, and it is not integrated with the analysis of eye-tracking data.	An integrated way to perform process and analyze think-aloud data (semi-)automatedly and link it with the mobile eye-tracking data.
Geographical context and locomotion	The context of the (visual and physical) behaviors and mental processes as the test person constantly interacts with the environment (Where do these behaviors take place?)	Location data is rarely recorded during the execution of experiments and not included in the analysis.	Recording of location data (e.g. as GPS measurements) and the integration of it in the analysis.

In light of these kinds of desired aspects in typical eye-tracking GI user studies, the following additional components are proposed and will be implemented in the prototype solution:

- Automated object-based fixation mapping
- Screen-recording video processing

- (Semi-)automated think-aloud data processing that includes transcription, segmentation, and encoding
- Synchronization of mobile eye-tracking data, think-aloud data, screen-content data, and GPS recordings

The automated object-based fixation mapping component should be able to process the scene camera video with the fixation data and identify the object of regard of every fixation. It should also be able to identify a wide range of objects that are of interest in GI user research (e.g. the cell phone as the mobile display). It is decided to map fixations instead of raw gaze points because the same is conducted during manual mapping (also in some automated mapping practice such as Wolf et al., 2018), and it significantly reduces the load of data processing.

Screen-recording video is chosen instead of activity logs because screen-recording is easier to acquire during an experiment. Action logs often need to be explicitly coded during the development of the mobile application, making them less easy to acquire. Actions logs are also very application-specific in terms of log items and log structure thus require tailored processing procedures. The recording and processing of screen-recording videos is a more generic procedure with an easier data-collection setup that could be applied in different experiment setups. The screen-recording processing component should be able to describe and identify the screen content at a given moment. The result of this component should be directly integrated into the result of the automated fixation mapping component.

The think-aloud data processing component follows the three typical steps in verbal protocol analysis: transcription, segmentation and encoding (Ericsson & Simon, 1993; Yang, 2003). The think-aloud audio will be processed into transcripts, protocol segments, and encoded protocols. The encoding should allow some degrees of flexibility of code definition (i.e. the researcher should be able to define coding schemes, instead of having to use a set of codes pre-defined by the prototype). The (intermediate) result of the process should also have an easy-to-access structure (e.g. with timestamps) that will allow it to be synchronized with other (processed) data. The results should also have a readable structure for humans in case they are qualitatively studied.

Each data-processing component should be able to produce results independently, but their results should also have a structure that allows them to be synchronized and integrated with others. The synchronization component provides an opportunities to bring together the processed data for integrated analysis. Commonly used metrics and visualizations (e.g. discussed in Chapter 2) can be generated with the individual or integrated datasets.

A graph illustrating the relationship between the source data, the proposed components and the desired information to be derived is shown in Figure 4-1.

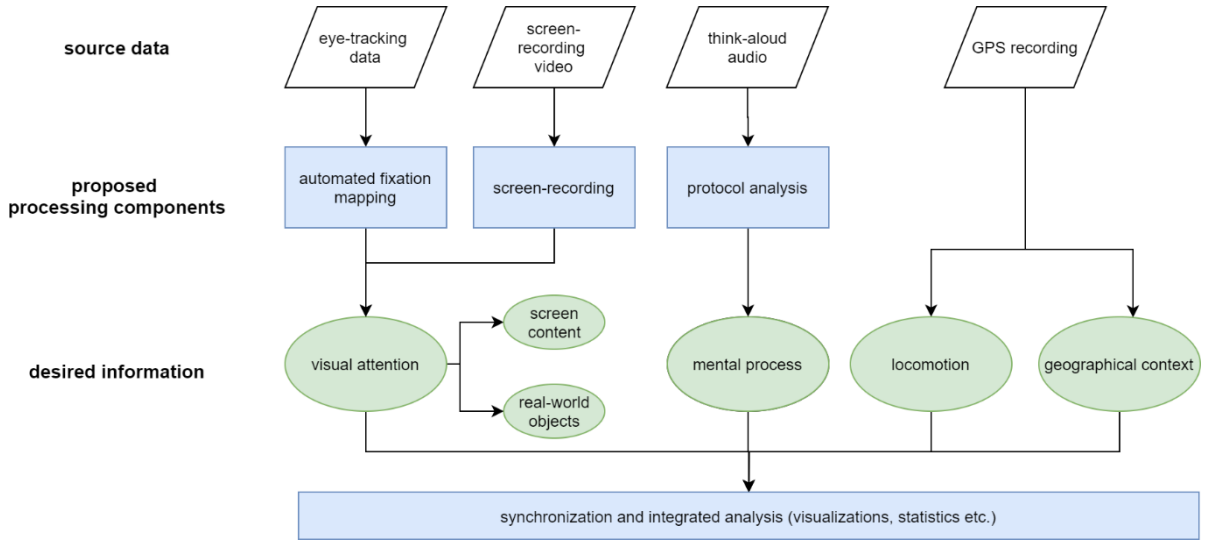


Figure 4-1 Requirements: deriving desired information with proposed components.

4.3. Implementation

4.3.1. Implementation framework

The implementation mainly consists of a fixation mapping component, a screen-recording processing component, and a think-aloud data processing component. The result from the components, together with GPS recordings, can be synchronized and analyzed together if needed. The implementation framework is shown in Figure 4-2. The prototype is implemented in Python as command-line tools. The following sections will explain the implementation details of each component. Information on the source code repositories and instructions to run the prototype solution is provided in Appendix A. For this first-stage prototype, the implementation is based on and limited to Tobii's hard- and software (data structure).

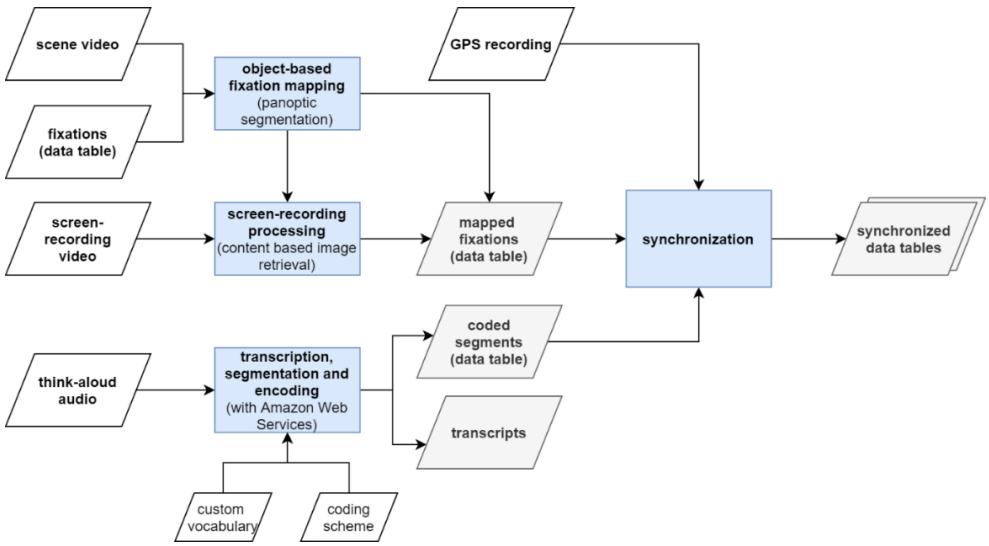


Figure 4-2 Implementation framework

4.3.2. Object-based fixation mapping

The task of the fixation mapping component can be described as: given an image (i.e., a frame from the scene video) and a pair of pixel coordinates (i.e., the fixation on the frame), find the object in the image

that this pixel belongs to. This task can be supported by image segmentation models. The implementation workflow is shown in Figure 4-3.

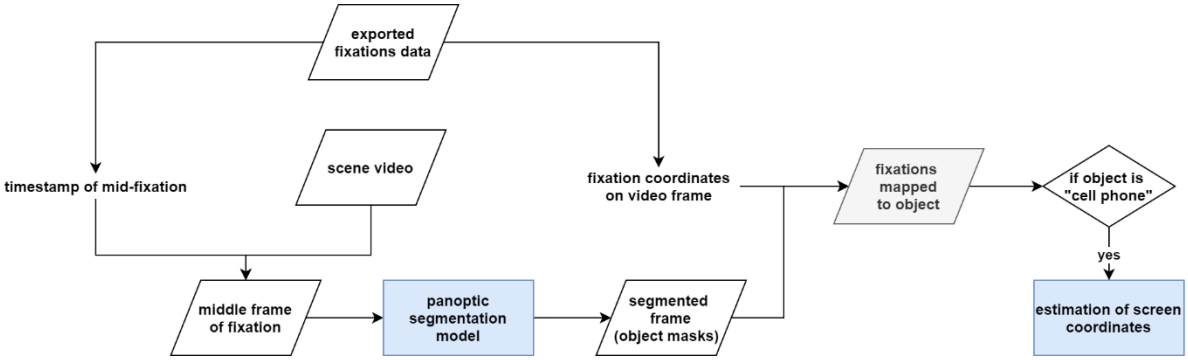


Figure 4-3 Workflow for mapping fixations to real-world objects

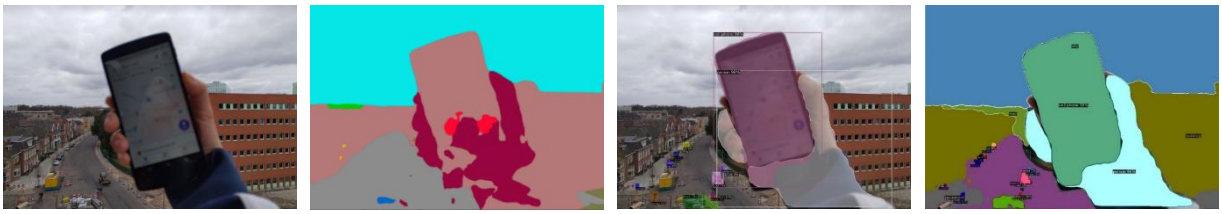
The automated mapping is performed on pre-filtered fixations exported from Tobii Pro Lab. During the mapping, each fixation is represented by its middle frame (i.e., the video frame closest to the middle timestamp of the fixation, similar to the procedure taken by Wolf et al., 2018) The middle frame is extracted from the scene video recording using the timestamp information of the fixation. The frame image is then segmented by the panoptic segmentation model, and the fixation is mapped to the objects given the object masks and the fixation coordinates. If the fixation is mapped to object "cell phone" (the mobile display), the screen coordinates of the fixation is then estimated. Since the prototype solution is built upon Tobii's services and file formats, the inputs are required to follow the Tobii data export file format.

4.3.2.1. Panoptic segmentation and the trained model

A pre-trained panoptic segmentation model from the Detectron2 platform is adopted for the fixation mapping task. Detectron2 is a platform powered by the PyTorch deep learning framework that implements state-of-art object detection algorithms and provides flexible customization to the models (Wu, Kirillov, Massa, Lo, & Girshick, 2019). The model is pre-trained with the COCO (Common Object in Context) panoptic dataset (Caesar, Uijlings, & Ferrari, 2016). Detailed configurations of the model are provided in Appendix B.

A panoptic segmentation model is chosen for the fixation mapping task because of its ability to coherently segment the entire scene, which enables it to address the most interesting objects (e.g. the mobile display, buildings, backgrounds) at the same time in a coherent manner. In image segmentation, the objects are generally grouped into two major categories: things (countable objects with well-defined shapes), and stuff (amorphous regions such as sky or grass; see Caesar et al. [2016] for the discussion of thing and stuff in segmentation tasks). In the application of mobile eye-tracking in GI user research, the objects of interest include both things and stuff. For example, the mobile display (cell phone or tablet) is a countable object (thing), while many objects in the environment such as buildings and roads are amorphous regions (stuff). Traditionally, these two categories are treated with different segmentation tasks: stuff is addressed with semantic segmentation where each *pixel* in the image is assigned a class label (semantic segmentation treats thing classes as stuff); things are typically addressed with instance segmentation where each *object instance* is detected and delineated with a segmentation mask. Although they seem related, these two tasks are very different. They are performed with different models, datasets, evaluation metrics, etc., and they cannot be directly integrated to achieve a complete and harmonious segmentation of an entire image. Instance segmentation cannot address stuff classes; semantic segmentation models perform well on bigger stuff

(e.g. sky), but tend to perform poorly on thing classes (Zhou et al., 2019). Panoptic segmentation is a recently proposed task that unifies the two traditional segmentation tasks (i.e., semantic segmentation and instance segmentation) by assigning each image pixel a semantic (class) label, as well as an instance label if a countable object is detected, providing a way to coherently segment the entire image (Kirillov, He, Girshick, Rother, & Dollár, 2018). An example of an image and the segmentation results from a semantic segmentation model, an instance segmentation model, and a panoptic segmentation model is shown in Figure 4-4. The example image (a) shows a typical scene where a map is shown on a cell phone in an outdoor environment with buildings, traffic, and other surrounding objects. The semantic segmentation model (b) is relatively good at segmenting big amorphous regions such as sky and buildings but performed poorly with countable objects such as person (the hand) and cell phone. The instance segmentation model (c) can detect objects and delineate object instances masks very well but cannot address the amorphous regions. The panoptic segmentation model (d) addresses both categories harmoniously with reasonable performance.



a) b) c) d)

Figure 4-4 An example of different segmentation models. a) the original image; b) semantic segmentation with UperNet101 network trained on the ADE20K dataset (UperNet101: Xiao, Liu, Zhou, Jiang, & Sun, 2018; ADE20K: Zhou et al., 2016), c) instance segmentation with Mask R-CNN trained on COCO dataset (He et al., 2017); d) panoptic segmentation with panoptic FCN trained on COCO panoptic dataset.

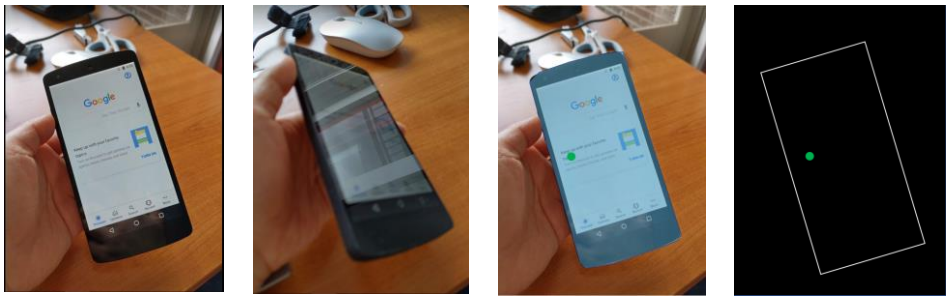
A pre-trained model is preferred against training a new model because the prototype solution aims for a more generic application within the scope of GI science, and the pre-trained model already covers a large number of categories that are also common for such applications. Training a model from scratch will require a huge number of labeled data and is extremely time and resource consuming. Re-training the pre-trained model with a new set of segmentation targets can make the model specialize on those targets; re-training is only needed in very case-specific circumstances when the target objects are relatively uncommon and not included in the pre-train model (such as object "syringe" in Wolf et al. 2018). The COCO dataset is one of the widely-used annotated image datasets that are used to train and validate segmentation models. The COCO panoptic dataset has 80 thing categories and 91 stuff categories. It covers many common objects that are of interest in the application of GI user research (such as cell phone, building and various building parts, indoor landmarks). A list of categories in the COCO panoptic dataset is provided in Appendix C.

4.3.2.2. Estimation of screen-coordinates of fixations on mobile display

The mobile display itself is often an object of particular interest. Together with information from screen-recording videos (Section 4.3.3), estimating the screen-coordinates of fixations might help to determine what is being looked at on the mobile display. However, it is important to notice that due to the precision and accuracy of the mobile eye-tracker itself, the calibration procedure of the experiments, and the assumptions made during the estimation, the estimated screen-coordinates are approximates that cannot be treated as an equivalent to the screen-coordinates of fixations in screen-based eye-tracking studies.

The estimation of screen-coordinates is based on two basic assumptions. Firstly, it is assumed that when the fixation lands on the screen, the entire screen is in the field of view of the scene camera. Secondly, when the fixation lands on the screen, the device is held in a relatively "upright" position where the screen

is relatively perpendicular to the line of sight (Figure 4-5a). From visually examining the video from the case study, the first assumption is likely to hold for smaller devices such as mobile phones, but might not be true for larger tablets. The second assumption also generally holds for smaller devices because the device is often needed to be held in a relatively upright position for the user to be able to read the screen content (Figure 4-5b). These assumptions, as well as the errors that they might introduce, will be further discussed later in the evaluation (Section 6.2.2).



a) b) c) d)

Figure 4-5 Estimating screen-coordinates of fixations on the mobile display. a) assumption of the phone in "upright" position, content is clear to read; b) phone in a tilted position, content is difficult to read; c) fixation (green dot) on object mask for the "cell phone" instance; d) a rotated bounding box is calculated for the mask to estimate the screen-coordinates of the fixation

Once a fixation is mapped to the category "cell phone," it is passed on to the coordinates-estimation unit. The object mask of the cell phone is extracted, where a rotated bounding box (minimal bounding rectangle) is calculated based on the mask. The fixation point is then used to calculate relative distances to the edges of the bounding box to get relative/proportional coordinates. With additional knowledge of the size of the device, absolute screen-coordinates estimations can be calculated. An example of this is shown in Figure 4-5 (c and d).

4.3.3. Screen-recording processing

In an ideal experiment setup, the screen-recording video is made simultaneously with the eye-tracking video. Once the videos are synchronized (see section 4.3.5 on synchronization procedure), every fixation that lands on the phone can be further mapped to the content of the phone (i.e., a frame from the screen-recording video) at that moment.

The goal of the screen-recording processing component is to determine the screen content at a given moment. The screen-recording processing component performs a content-based image retrieval (CBIR) task: given a query image, it finds the most similar image from a pool of (pre-defined) candidate images (Figure 4-6). The task is executed in two phases: the indexing phase and the search phase. In the indexing phase, representative frames (e.g. screen contents of particular interest) from the screen-recording video are manually selected as candidates. They are passed to a feature descriptor, where their feature vectors are calculated, indexed and stored in a file. This indexing operation is independent of the eye-tracking data. During the search phase, for each fixation mapped to the cell phone, a frame from in the screen-recording video is extracted using the timestamp of the fixation as the query image. The same set of features of the query image is also calculated with the descriptor, and the feature vector of the query image is compared to those of the candidate frames with a distance metric (in this case, the Euclidean distance). The best-matching candidate with the smallest distance is assigned as the screen content of the fixation.

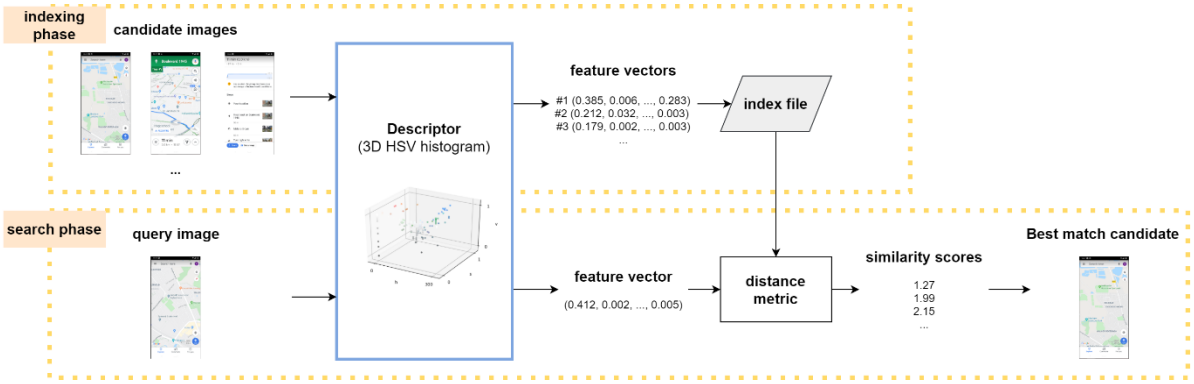


Figure 4-6 Content-based image retrieval for screen-recording processing

The descriptor used in the prototype solution is (3D) HSV histograms. Because screen-recording videos are not subjected to changes in lighting conditions or camera angles, color histograms are able to capture image features stably without too much noise. Compared to the RGB color space, the HSV color space is more similar to the human perception of color and is relatively computationally simple compared to sophisticated systems such as the CEILab color space. The underlying assumption of using color histograms as the descriptor is that images with similar color distributions are considered to have similar contents. Because the mobile display often consists of regular-shaped sections (e.g. the instruction section and map section in Ohm et al., 2017) with different content and color distributions, instead of calculating one histogram for the whole image, the image is divided into four sections: upper left, upper right, lower left, and lower right, and histograms are calculated for every section. The resulting feature vectors contain histogram information for all four sections. The bin number of the histograms determines the sensitivity of the comparison. The more bins, the more detailed the histograms are, the more detailed the comparison and matching can be. But a larger number of bins also significantly increases computation time. Determining the number of bins will depend on the perceived similarity among the candidate images, and the expected level of detail in the comparison. Tuning the bin number is an experimental and iterative process. The prototype script allows the user to define the number of bins for each of the H, S, V channels, but it offers a default number for a relatively detailed comparison.

4.3.4. Think-aloud processing

In this processing component, think-aloud audio data is processed with a semi-automated transcription – segmentation – encoding workflow enabled by Amazon Web Services (AWS). Apart from the audio data input, a list of vocabularies should be provided for the transcription, and a coding scheme with example utterances should be provided for the encoding. Transcripts, segments and coded protocols are produced as output. Figure 4-7 shows the workflow. The processing procedure can be performed in the AWS web console by manually configuring the various services or from scripts that chain the services as a cloud workflow (Section 4.3.4.2).

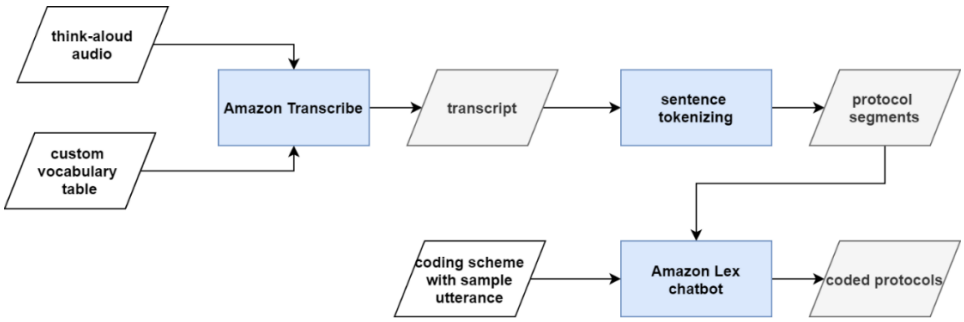


Figure 4-7 Workflow for think-aloud audio processing

4.3.4.1. Transcription and segmentation

Audio data is transcribed with Amazon Transcribe (Amazon Web Services, 2020c), an automated service that performs speech-to-text tasks. Transcripts are returned as JSON files that consist of both full transcripts and precise timestamps for each distinguishable word. The full transcript is easily readable for the human, and the word timestamps offers convenience for further processing of the transcripts and synchronization with eye-tracking data. A drawback here in comparison to manual transcription is that speaking accents can significantly influence the quality of the transcription. Audios with strong accents cannot be accurately transcribed.

Custom vocabularies are used to enhance automated transcription. Custom vocabulary, often with custom pronunciation definitions, is a list of words (i.e., often domain-specific or proper nouns) that can be defined to help improve the quality of the transcript (Amazon Web Services, 2020c). Defining custom vocabulary is often necessary when proper nouns such as place names are present in the audio and when these words are of particular interest, which is often the case in GI user studies.

Transcriptions are then segmented into sentences by sentence tokenizing. The segments should contain enough information to be allocated with a code (Ericsson & Simon, 1993), and the length of sentences is considered sufficient. Sentence-level segmentation is also used in GI user research such as C. Wang et al. (2019) and Viaene, Vanclooster, Ooms, & De Maeyer (2015). The timestamp information is preserved during the segmentation.

4.3.4.2. Encoding

The encoding unit in the prototype assigns codes to protocols based on a custom coding scheme using a chatbot service. The encoding of think-aloud segments can be regarded as a text classification task. The conventional solutions for such classification problems often involve complex machine learning models performing natural language understanding (NLU) tasks. Such solutions require training of a deep learning model on a huge amount of labeled data with massive demands on time and infrastructure. Implementing or training such models only for the encoding of think-aloud protocols, which are often of a much small amount comparing to the data needed to train the model, is not efficient. And because there are no unified coding schemes for think-aloud protocols, implementing and training one model with one coding scheme that can only be used in one study is not efficient either. To make the encoding process more automated and more general, an alternative is proposed using existing chatbot services.

Chatbots are intended to offer reasonable and engaging human conversations through text or audio. Web service providers such as Amazon now offers customizable, deployable chatbots as services. These chatbot services are powered by pre-trained NLP and NLU models and transfer learning technologies at the backend, so they can learn to predict new classes faster with much fewer training samples (Metallinou, 2018). The job of a chatbot can be simplified as detecting the "intent" of an input utterance and providing a pre-defined response to it. An intent is the category of the meaning of an utterance. For example, both utterances of "how should I go to building X?" and "what's the route to building X?" have the intent of "asking for direction." The chatbot returns a response once it recognizes an intent in the input utterance. It can return a piece of pre-defined text or perform an operation. If it cannot recognize the intent in the input utterance, the pre-defined "fallback" intent is activated and the chatbot will return the pre-defined fallback response. Chatbot services allow users to create customized intents with sample utterances and customized responses.

In the case of coding think-aloud protocols, each protocol segment is an input utterance, and the coding scheme are the intents. For each code, a number of sample utterances are given to train the chatbot. After

the chatbot is trained, protocol segments are passed as input and codes are allocated via the responses. Although the segments are not coded in a random order, because the chatbot reacts to each input independently, the encoding of the segments is independent and thus can be considered as "context-free" (Ericsson & Simon, 1993). Because intents cannot overlap, each sentence is only allowed to be allocated with one code. To deal with protocols that are irrelevant to the analysis (i.e., irrelevant to the coding scheme), a fallback intent and a corresponding response are defined so that the chatbot returns "unclassified" when it cannot recognize the intent of a given sentence.

The encoding unit is implemented with Amazon Lex chatbot service (Amazon Web Services, 2020b). It takes protocol segments as input and returns these segments with codes allocated to them according to the coding scheme. To prepare for the encoding, the coding scheme is defined and sample utterances under each code category are developed. Then a customized chatbot is built using the coding scheme and the sample utterances. After that, segments can be passed into the chatbot and codes can be allocated. The timestamp information (start and end timestamps for each sentence) of the protocol segments is preserved in the output.

The major advantage of using a chatbot service for encoding is the low cost of training, which enables the classification targets to be very flexible, where researchers can define their own coding scheme easily. The amount of sample utterances needed to train an Amazon Lex chatbot is significantly smaller than the number of labeled sentences needed to train a text classification model, and the training time is significantly lower. The quotas for sample utterances per intent is 1500 in Amazon Lex, but normally only a few dozen are provided in practice (e.g. the official Lex examples have less than 10 utterances), and it is recommended that using fewer utterances may increase Lex's ability to better recognize inputs outside the provided utterance set (Amazon Web Services, 2020e). However, since the chatbot service is offered in the form of a black box, it's not possible to fine-tune parameters or customize the backend model. Monitoring training and prediction performance are also less systematic.

4.3.4.3. A cloud workflow

The procedures described in the previous sections can be configured manually in the AWS web console of the corresponding services. To perform batch processing of think-aloud audios without manually configuring every step the console, a workflow enabled by Amazon's cloud infrastructure and services is implemented with the AWS Boto3 API. The workflow consists of 3 scripts: transcription, bot-building and encoding. The workflow and its infrastructure are shown in Figure 4-8. It utilizes AWS's storage (S3), transcription (Transcribe), chatbot (Lex), serverless computing (Lambda) and access management services (IAM). Input audios are first stored in an S3 bucket. The transcription script starts Transcribe jobs for the audio files and stores the output transcripts to an S3 bucket. The storage of a transcription file triggers a Lambda function that tokenizes the transcript into sentence segments (this operation is not performed locally, but on Amazon's server). The Lex chatbot is built with the build_bot script. The encoding script will iterate through the S3 bucket and encode every segment file by invoking the encoding Lambda function that passes sentences to the Lex chatbot and retrieves responses, the coded protocols are saved back into the S3 bucket. IAM service is involved throughout the workflow to manage access among the services (e.g. reading and writing into S3 buckets, invoking Lambda).

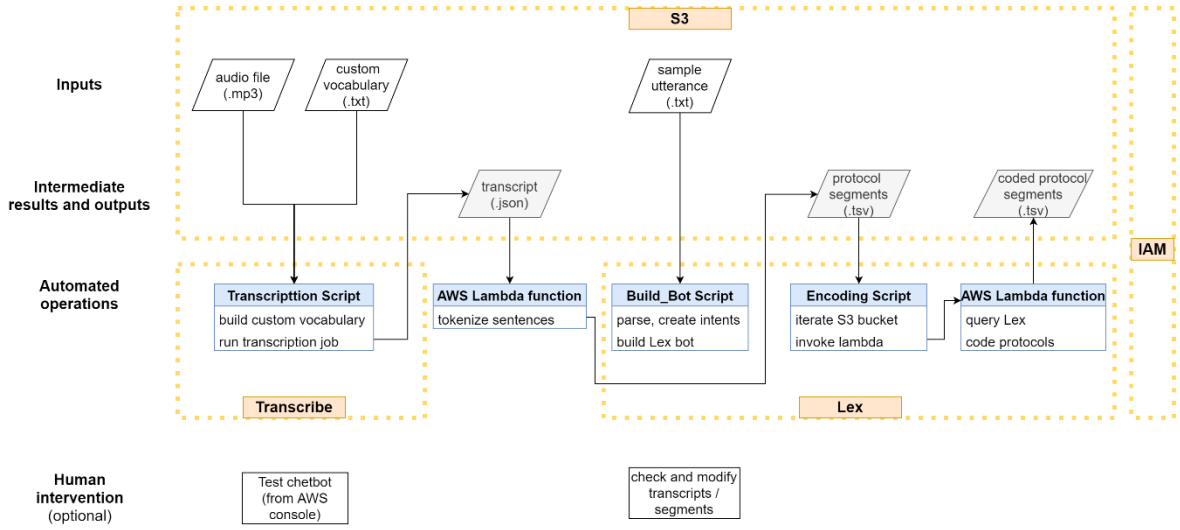


Figure 4-8 Cloud workflow for think-aloud data processing with AWS

Although it's technically possible to chain the entire workflow in one script, it is decided to have three separate scripts to allow space for human intervention in between the steps to enable supervision and modification. After the automated transcriptions, transcripts can be checked for quality so that the custom vocabularies can be modified if certain words are not successfully identified. After bot-building and before passing all protocol segments to encode, the Lex chatbot can also be tested and re-trained (in the AWS web console) until the responses are satisfactory to code all the protocol segments.

4.3.5. Synchronization and integrated analysis

4.3.5.1. Synchronization

The results from the previous processing components, together with the GPS recording data, are synchronized based on their timestamps. The goal of the synchronization is not to have all the data in one data table, but to provide a possibility to bring different datasets together to present different aspects of the interaction process between the environment, the mobile display and the mental map. And the result will not be one unified data table with all the data from different sources. Instead, multiple synchronized data tables can be produced. The synchronized data tables can be used to calculate metrics for quantitative analysis, it can also be visualized to support exploratory analysis. The synchronization is based on the temporal characteristics of the datasets (Table 4-2).

Table 4-2 Temporal characteristics of the datasets

Data source	Timestamp precision	Duration of an event
Scene video	1ms	40ms*
Screen-recording video	1ms	20-40ms**
Mapped fixations	1ms	~50ms- ~1s
Think-aloud protocol segments	0.01s	Several seconds
GPS recording	1s	N/A

*Scene videos are recorded in 25fps with Tobii Pro Glasses 2, might vary for other hardware. **It can an vary depending on the configuration of the screen-recording.

To synchronize eye-tracking data (scene video and fixation data) with screen-recording video in order to map fixations to screen contents, a time offset (i.e., the difference between the start of the screen-recording video and the eye-tracking recording) is needed. This time offset can be manually determined by

manually inspecting the scene video to locate the time when the screen-recording started. It can also be directly extracted from the eye-tracking data if the Tobii sync port is used during the experiment, the sync port is a hardware feature in Tobii Pro Glass 2, where a TTL 3.3V signal can be communicated between the Tobii recording unit and an external device, the signal is registered as an event in the eye-tracking data. Details of the sync ports can be found in the user manual (Tobii Pro, 2016).

Because think-aloud protocols normally have a duration of several seconds, multiple fixation events can take place during the time span of one protocol segment. To associate fixations with protocols, all fixations with the middle timestamp that fall between the time interval of the protocol segment are assigned to that segment. Because audio is simultaneously recorded with the eye-tracking video by the eye-tracker, no time offset is needed here.

Mapped fixations can be synchronized with GPS recordings by assigning the most frequent fixation target to each GPS point. Fixation are assigned to a GPS point when their middle timestamp falls within the GPS measurement interval. The most frequent fixation target among the fixations assigned to one GPS point can be used to represent the target of the visual attention at that location. A time offset is needed to determine the time difference between the start of the GPS timestamp and the eye-tracking timestamp. The offset can be determined by manually inspecting the video or automatically using the sync port.

GPS points are assigned to think-aloud protocol segments when the timestamp of the measurement falls between the time interval of the protocol segment. Each think-aloud protocol segment can correspond to multiple GPS points.

4.3.5.2. Some possibilities for successive analysis

The resulting synchronized data tables might be used for exploratory or quantitative analysis. The actual analysis will depend on the specific research questions and study design. This subsection presents some possible analysis that can be carried out with the data produced in previous steps.

After fixations are mapped to objects and screen-contents, eye-movement metrics such as fixation count, duration, mean fixation duration can be calculated, visualized, and used for statistical analysis, and sequence of the visual attention can be qualitatively explored with visualizations (Figure 4-9; Göbel et al., 2019) or quantitatively compared with sequence analysis (e.g. the sequence analysis for screen-based eye-tracking in Çöltekin, Fabrikant, & Lacayo, 2010). The mapped fixations might also be used for data-driven analysis such as the inference of user tasks (as suggested by Liao, Dong, Huang, Gartner, & Liu, 2019).

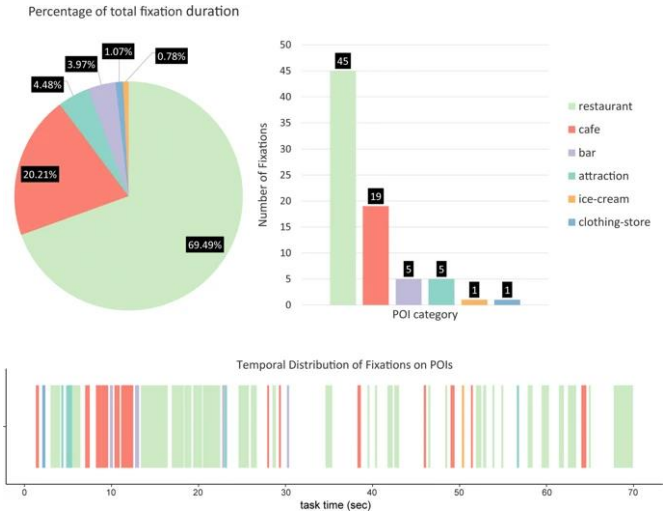


Figure 4-9 An example of visualizing the distribution and sequence of fixation (source: Göbel et al., 2019)

The processed think-aloud protocols (transcripts, segments, coded protocols) might be used for protocol analysis (e.g. Viaene et al., 2015). When synchronized with mapped fixations, the relationship between verbalizations and eye-movements can be explored (e.g. by calculating the percentage objects mentioned in verbalization segments that are also fixated on, as in Viaene, Ooms, Vansteenkiste, Lenoir, & De Maeyer, 2014). It can also be used to assist exploratory purposes (e.g. as synchronized exploration with fixation data).

When GPS data is synchronized with mapped fixations or think-aloud protocols, the spatial distribution of visual attention or utterances along the route can be jointly explored (for example, using heatmaps to reveal the spatial distribution of map usage as in Kiefer et al., 2011, Figure 4-10); and visual attention can be associated with locomotion (e.g. relating map reading with walking speed in Kiefer et al., 2011). Spatial analysis such as viewshed analysis can also be possible given a 3D data model of the (urban) environment (e.g. OSM buildings).

Figure 4-10 An example of exploring the spatial distribution of visual attention: point density heatmap to highlight the usage of the map along the route, red color indicates high accumulated map usage time (Source: Kiefer et al., 2011)

4.4. Summary

This chapter presented the design and implementation of the prototype solution. Based on the discussion in Chapter 2, the requirements for the analytical solutions were formulated to acquire the desired information on visual attention, mental processes and geographical context. In the implemented prototype solution, the fixation mapping component used panoptic segmentation to map fixations to real-world objects in the environments, it also estimated the screen-coordinates for fixations landing on the mobile display; the screen-recording processing component used histogram matching to help associate fixations with screen contents on the mobile display; the think-aloud data processing component transcribed, segmented, and encoded think-aloud protocols in a cloud workflow enabled by Amazon Web Services; the result from these components, with location data (GPS recordings), could be synchronized and analyzed together. The next chapter will demonstrate the functionalities of the prototype solution with the GeoFARA case study.

5. DEMONSTRATION: THE GEOFARA CASE STUDY

5.1. Introduction

This chapter demonstrates the functionalities of the prototype solution with a case study. The proof-of-concept case study performs an exploratory analysis with the data collected for the evaluation study of GeoFARA. It will try to answer the questions about the use and usability of GeoFARA with a combination of mobile eye-tracking, screen-recording, think-aloud and location data.

5.2. Data

The GeoFARA evaluation session was conducted with two pilot test persons and 14 formal test persons. The recordings of the two pilot tests of the GeoFARA evaluation study were used in the demonstration. They were chosen because they had relatively good think-aloud data. There was too much silence in the recordings of the actual test persons, and the actual tests persons had stronger accents which made transcription difficult. Also, doing the simulation for data not collected on-site (described later in this section) was only feasible for a very limited number of recordings. The recording of pilot A was 65 minutes in length with 75% overall gaze samples (percentage of correctly identified gazes); the recording of pilot B was 31 minutes in length with 74% overall gaze samples.

Because the fieldwork area was relatively large, it was divided into five sub-areas (Figure 5-1). The recordings were also split into scenes, and each scene covered part of the fieldwork area (Table 5-1). Originally a total of 8 scenes were included, but due to low (local) sampling rate, part of the recordings (approx. 12 minutes in length, with approx. 50% sampling rate) from Pilot B was discarded. The five scenes from pilot A were in sequential order. The approximate routes of the test persons are also shown in Figure 5-1.

Figure 5-1 Sub-areas and walking routes of the test persons in the case study. Walking routes were manually delineated according to video recording to give an impression of where the test persons had been to.

Table 5-1 Scenes of recordings in the case study

Scene	Pilot Person*	Duration (mm:ss)	Sampling rate
East	A	11:07	86%
Villa	A	09:32	82%
Store	A	16:13	64%
Wall	A	11:21	71%
Corner	A	14:28	81%
Middle	B	14:29	87%

*The following analysis in the chapter will only be based on scenes, and will not distinguish between the two pilot persons

Screen-recording and GPS data were not collected as part of the original data during the execution of the GeoFARA evaluation study. For the purpose of demonstrating the functionality of this prototype solution, the data was collected with simulations. For the screen-recording, the video recording was played on the computer, and the author replicated the test persons' interaction with GeoFARA by operating the phone following the test person in the video recordings. The content on the phone screen was recorded with a screen-recording app. Due to safety concerns (this highly-focused operation was not possible when walking in live traffic), the simulation was conducted indoor instead of outdoor walking in the original study area. The simulation was conducted for all six scenes. For the GPS recording, the author played the video recording and followed the same routes of the test persons with a mobile phone recording GPS measurements with a frequency of 1 per second. The simulation was conducted with four scenes, except the *villa* and the *wall* scenes, because the villa was not publicly accessible anymore of the time of the simulation, and the *wall* scene involved a road crossing with traffic lights, which made it impossible to replicate the route with the same pace as the test person.

5.3. Analyzing visual attention: real-world objects, screen contents and screen coordinates

The raw gaze data was filtered with the Tobii I-VT Attention filter and exported from Tobii Pro Lab. The fixations were then mapped to real-world objects with the panoptic segmentation model (as described in Section 4.3.2.1). The target objects were classified into four categories: cell phone, building, surroundings, and others. Because all POIs shown in GeoFARA were buildings or built-up objects, these objects were grouped to the building category. The surroundings category includes other common objects in the environments that are not buildings. The first three categories (i.e., phone, building and surroundings) should include most of the objects that might be present in the scenes. The category "others" would catch any unexpected predictions from the segmentation model (e.g. objects like "paper" or "suitcase" that are not expected to appear in this particular fieldwork context). A list of objects in each category is shown in Table 5-2. It should be noted that since buildings are segmented as amorphous areas, individual buildings could not be identified.

Table 5-2 Categories of real-world objects for fixation mapping

Category	Objects	Explanation
cell phone	cell phone	the mobile display
building	building, house, wall, window, ceiling, wall-brick, wall-stone, wall-wood, bridge	buildings, built up objects and parts of buildings
surroundings	tree, fence, sky, pavement, grass, dirt, rock, road, water, river, sand, person, bicycle, car, motorcycle, bus, train, truck, traffic light, stop sign, parking meter	objects in the environment that don't belong to buildings (e.g. traffic, road, nature etc.)

The distribution of the fixations were measured with total fixation count, total fixation duration and mean fixation duration. Figure 5-2 shows the total fixation count and duration on each object category, and Figure 5-3 shows the mean fixation duration on each object category.

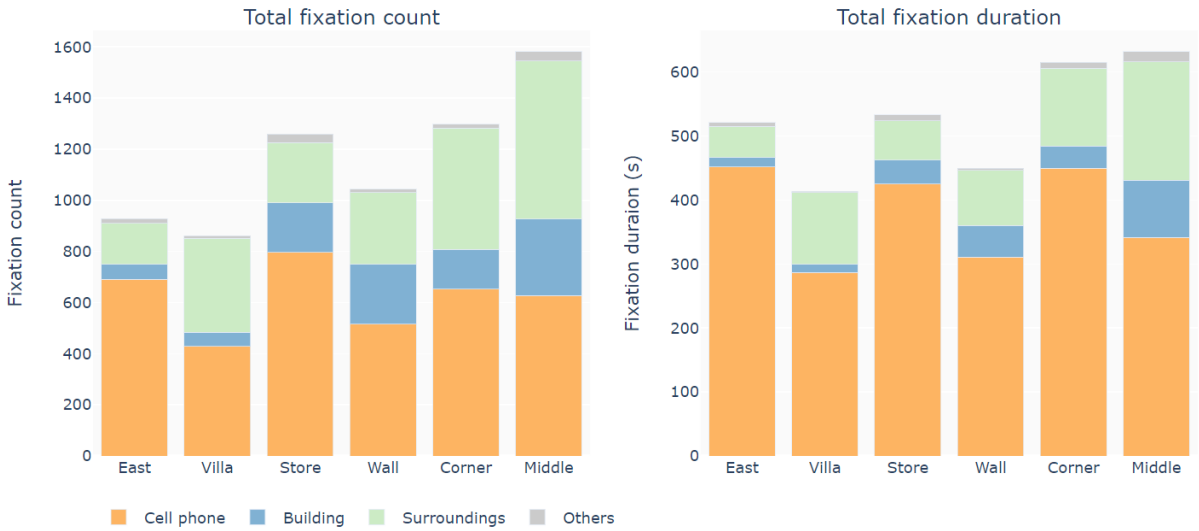


Figure 5-2 Distribution of fixation on object categories, total fixation count and total fixation duration

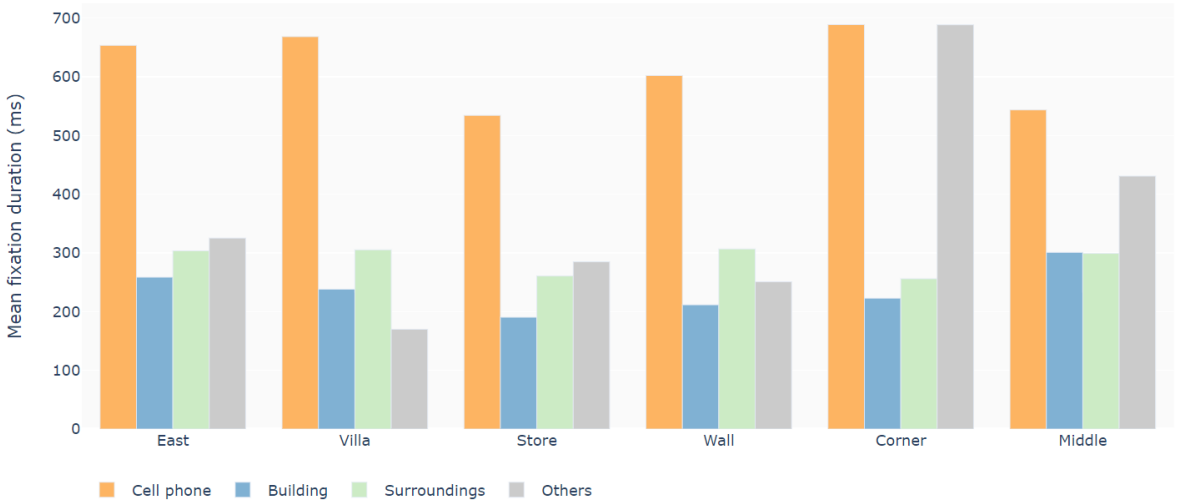
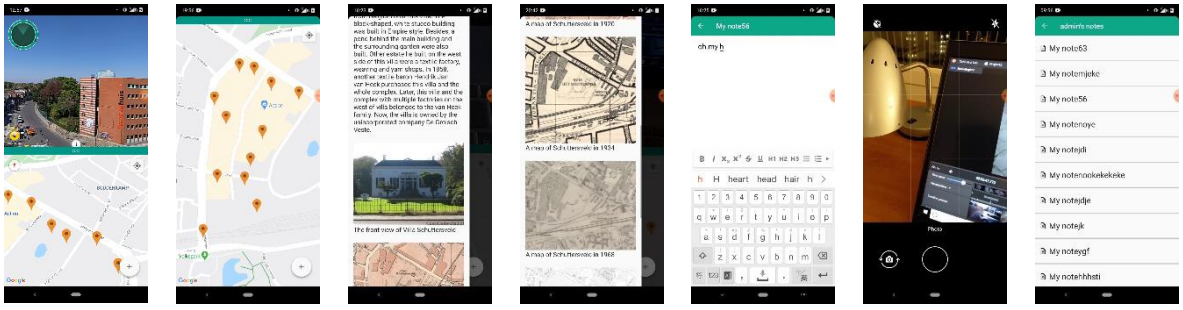


Figure 5-3 Mean fixation duration for object categories

For all six scenes, the test person paid much attention to the cell phone. Fixations on the cell phone take up nearly half of the total fixation count and over 60% of total fixation duration. It shows that the test persons generally relied on the app during the fieldwork. The surroundings are the second biggest target of attention, which could be explained by the attention on the ground and traffic during walking (Amati, Ghanbari Parmehr, McCarthy, & Sita, 2018). The distribution of fixations might be influenced by the environmental context. The villa area and the southern area had more trees, while the eastern, store, and wall areas were mainly built-up with fewer vegetations. It could result in relatively more attention on the surroundings (which includes trees) than buildings in scene *villa, corner*, and *middle*. The numbers of fixations classified to the "others" category are relatively low across the scenes, the percentage ranging from 1.08% to 2.26%, which indicates that most predicted objects were indeed included in the three main categories.

Mean fixation duration can indicate the processing demand of the stimuli (Gog, Kester, Nievelstein, Giesbers, & Paas, 2009). And important, interesting, or salient features can lead to longer fixations in scene perception (De Cock et al., 2019). In all six scenes, the mean fixation duration on the cell phone is relatively long. It shows that the test persons spent more effort in processing the information on the phone, possibly from reading the text or studying the map. The mean fixation duration on buildings and surroundings are similarly shorter. It indicates that the buildings were not perceived as more important or interesting than the surrounding environment. It could also indicate that the test persons had no difficulties recognizing the buildings. However, because individual buildings are not distinguishable, the results cannot show the difference in the attention spent on buildings that are remnants (POIs) and other building objects. This information can only be acquired with manual fixation mapping.

For each screen-recording video, representative screenshots were taken as candidate images. The candidate images were grouped into 7 categories: map-AR, full map, info, old map, take note, take photo, and view note. Figure 5-4 shows an example image for each category. The actual number and composition of candidate images for each scene can vary. For example, the candidate pool might consist of multiple screenshots of different info screens belonging to different POIs; and if the info was scrollable, multiple screenshots were taken; sometimes multiple screenshots were taken for the map-AR screen as well. Depending on the scene, not all seven categories were present in the candidate images.



Map-AR | Full map | Info | Old map | Take note | Take photo | View note

Figure 5-4 Example image for each category of screen-content

After the screen-recording video and the eye-tracking data were synchronized, the fixations on the cell phone could be mapped to screen contents (Section 4.3.3). Figure 5-5 shows the distribution of fixations on different screen contents in terms of total fixation count and total fixation duration, Figure 5-6 shows mean fixation durations on the screen contents.

Figure 5-5 Distribution of fixation on screen contents, total fixation count and total fixation duration

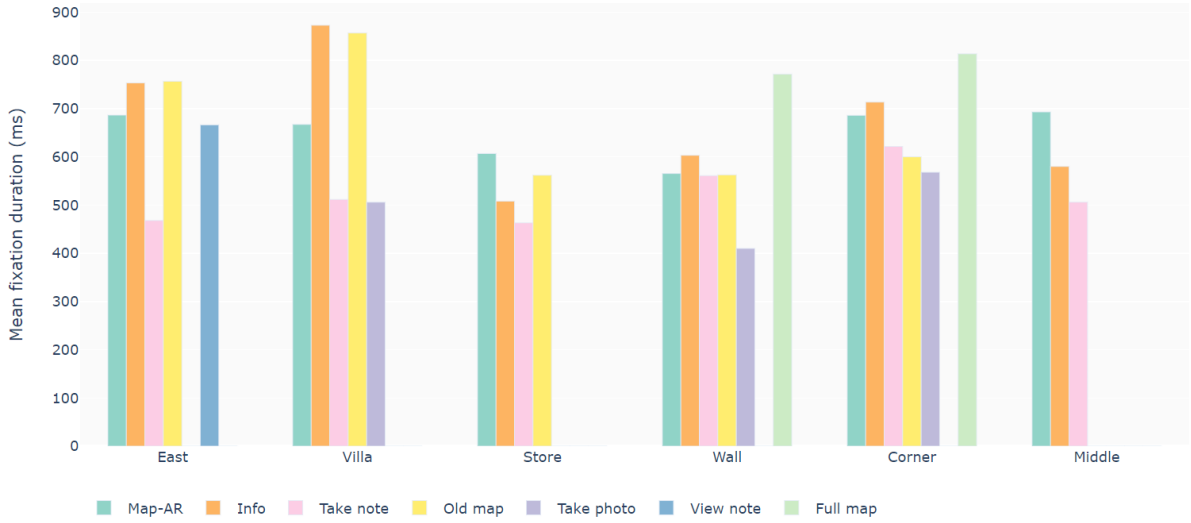


Figure 5-6 Mean fixation duration on screen contents

The distribution of fixations on different screen content can reflect the use of GeoFARA. Map-AR, info, and take-note screens were viewed in all six scenes. The test persons took time to read the information on the info screens. In the scene *east*, the test person spent more time on the map-AR screen, the reason might be that it was the very beginning of his fieldwork and he was not very familiar with the application, which could be confirmed by the recording video. In scene *middle*, the test person spent a large amount of time taking notes. Based on the recording, it was partly because he spent a lot of time trying to voice-type and correcting the voice input. The difference in mean fixation duration on screen contents is not as big as the difference between the phone and objects in the environment. The test persons generally had relatively long fixations on all screen contents. Among all the screen contents, the mean fixation durations on the map-AR screen and the info screen tended to be longer than others, which could be a result of studying the map and reading the text on the info screen. The mean fixations on the take-note screen tended to be shorter, which could be explained by the action of typing, with frequent switch of attention between the keyboard and the note content.

Because the first five scenes (*east, villa, store wall and corner*) were in temporal sequence, the data also provides insights on how this test person changed his use of the app during the fieldwork session. He

mostly relied on the map-AR screen for navigation at the beginning but also started to use the full-map screen towards the later part of the fieldwork. His mean fixation duration on the old maps was longer in the first two scenes than the later ones. It might be a result of getting familiar with the same set of old maps.

Switch frequency was calculated as an indicator of the interaction between the phone and the environment. A switch is defined as a change of fixation target. A switch of attention between the phone and the environment can be an indication of the test person connecting the information on the phone with the environment. Since buildings are of special interest, the number of switches between the phone and buildings were calculated separately from the number of switches between the phone and other objects. The number of switches was calculated per minute for the six scenes. The results are shown in Figure 5-7.

Figure 5-7 Switch count per minute between the phone and the environment

Generally, there are more switches between the phone and other objects than between the cell phone and buildings. While switches between cell phone and environment indicates more visual connections of information from the app and the environment and might indicate learning process, walking with a phone might lead to more switches in general, as compared to standing.

The pattern found in Figure 5-7 is further explored by plotting the fixation sequence on a timeline. The sequence includes both object categories and screen-contents. Figure 5-8 shows an example of the fixation sequence of minute 4 and 5 from scene *villa*. During minute 4 (180s to 240s), the test person spent continuous attention on the info and old maps (which resulted in a relatively lower number of switches between phone and environment). While in minute 5, he made more switches between the info screen and the surroundings. The recording video shows that during minute 4, the test person was standing in front of the Bistro building and reading the info screen; and during minute 5, he was mostly walking towards the next POI (Villa) while glancing the info screen.

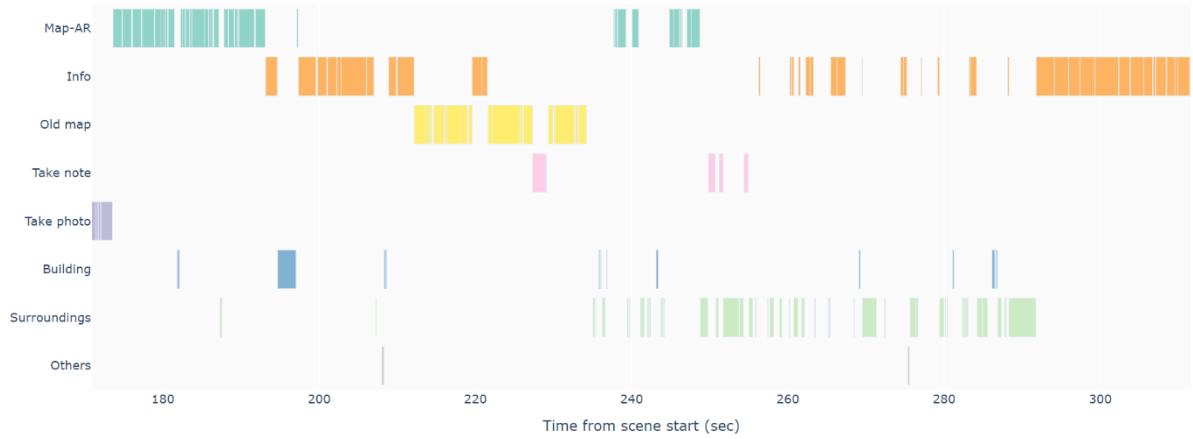


Figure 5-8 Fixation sequence: minute 4 and 5 from scene *villa*

Figure 5-9 shows the fixation sequence of minute 8 and 9 from scene *store*. During minute 8 (420s to 480s), the test person mainly looked at the old maps, and made a few interactions with the building, but there were not many interactions between the phone and the environment in general; during minute 9 (480s to 540s), the switches first took place between the phone (map-AR and info screen) and the surrounding, then between the phone (info screen) and building. The recording shows that during minute 8, the test person was mostly standing still, studying the old maps and wondering if the building Mason Manon was built on former factories. During minute 9, he first walked to the next POI (Leen Bakker) and stood in front of the building to read the info screen, comparing the building with the textile store described on the info screen.

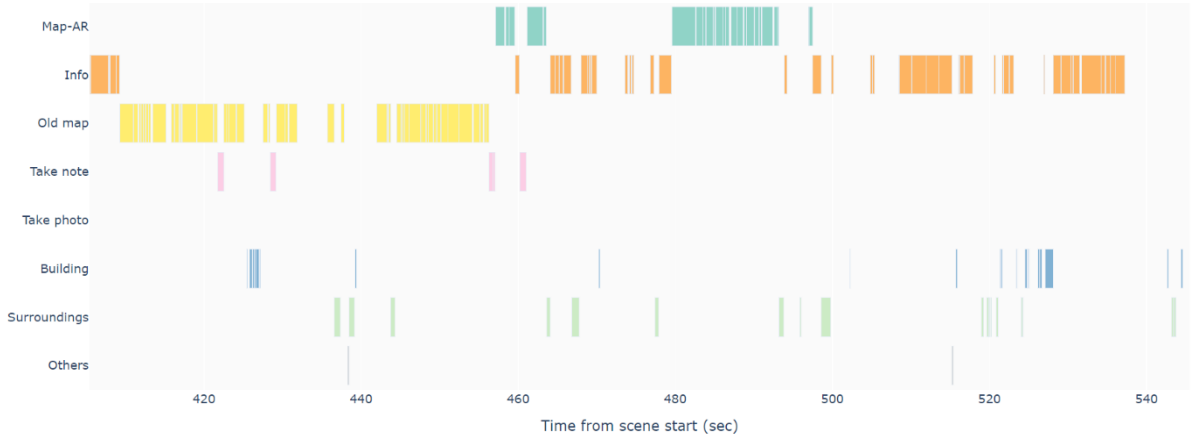
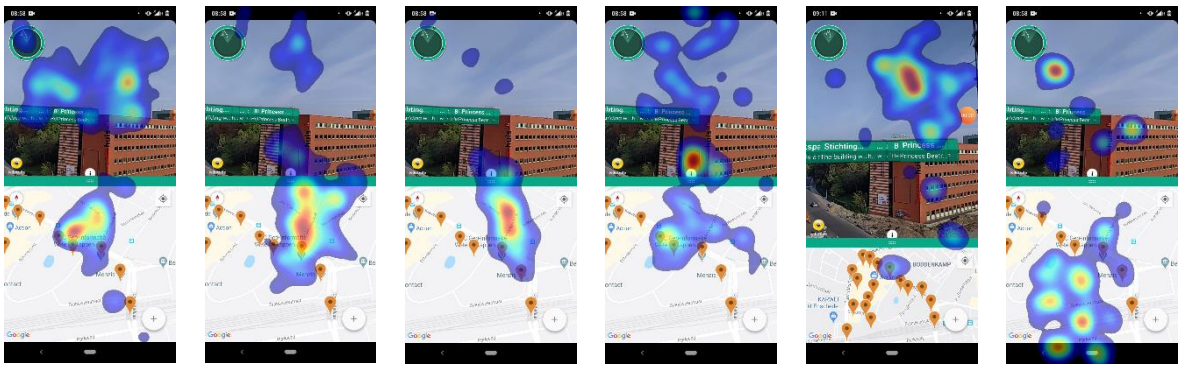


Figure 5-9 Fixation sequence: minute 8 and 9 from scene *store*

The exploration with other scenes shows similar results. Walking tends to result in more switches between the phone and the environment. These switches are mainly between the map-AR screen and the surroundings (when the test person used the map/AR for navigation), and between the info screen and the surroundings (when the test person did not need navigation). More switches between the phone and the building are often associated with learning about a particular building with the info or old maps. During the learning, there is often continuous attention on the phone, fixations on the phone are longer, and there are fewer switches between the phone and the environment. In these scenes fewer overall phone-environment switches but relatively more phone-building switches might be an indicator of the "learning" process. However, because the prototype cannot distinguish between individual buildings, it is not known *which* building the test person was looking at a certain moment. Context knowledge and manual

inspection of the recordings are needed to make interpretations of the patterns found in the fixation sequences.

The map-AR split-screen is one of the most important features of GeoFARA, it shows the POIs with an overview on the map and a live view through the AR at the same time. To investigate how the test persons use the map-AR split-screen, heatmaps were made using the screen coordinates of the fixations mapped to the map-AR screen. Because the actual content of the map-AR screen changes throughout the recording, an example image of the map-AR screen was chosen as the background image for all the heatmaps (Figure 5-10). (The test person changed the layout of the map-AR screen in scene *corner*, thus the background image for the heatmap was changed accordingly.) The heatmap are intended to give a rough indication of which part of the screen was viewed more, but not on detailed screen features (e.g. the labels in the AR or the icons on the map). So it was not known if the test person looked at a particular label or icon before opening the info tab. If this information is needed, it might be collected by logging of app interactions.



East Villa Store Wall Corner Middle

Figure 5-10 Heatmaps on map-AR screen. Because the actual content of the map-AR screen changes throughout the recording, a sample image of the map-AR screen was chosen as the background image for all the heatmaps.

The heatmaps show that the test persons utilized the AR as well as the map in all scenes. But in scene *villa, store* and *wall*, the map was used more than the AR. In scene *corner*, the AR part of the screen was expanded and the heatmap shows a lot of attention on the AR part of the screen. The recording video showed that the test person decided to use the AR more ("I'm gonna try to do that and look through [AR] just to see if that's more helpful"), and he also relied on the AR to navigate to the POI "Stichting 55+" when the map didn't give enough navigational assistant ("so this is an example where the augmented was helpful because I didn't have enough base map context").

5.4. Processing think-aloud protocols: identifying usability issues

This section demonstrates the procedure of using think-aloud protocols to discover the usability issues of GeoFARA. Using think-aloud protocols to explain visual behaviors is discussed in the next section.

The think-aloud audios were first transcribed and segmented (Section 4.3.4). A list of custom vocabularies was created to assist the transcription. The list consists of POI names and other words of interest to the task is shown in Table 5-3.

Table 5-3 Custom vocabularies

Category	Vocabulary
POI names	ITC, Menzis, Bistro, Het Koesthuis, Schuttersveld, Leen Bakker, Princess Beatrix, KPN, Maison-Manan, Kwantum, POCO, Praxis, Toy-champ, Fit For Free, Volkspark, Stichting,
Other words of interest	textile, remnant, factory, Enschede, villa, wall, van Heek, tunnel

After the audios were transcribed and the protocols were tokenized into sentence segments, an Amazon Lex chatbot was built with four intents to encode the protocols (as described in Section 4.3.4). The coding scheme is shown in Table 5-4. The coding scheme was developed by basic themes that were expected in the protocols. Four in-scope codes (I – app interaction, M – movement and navigation; T – task-related and Y – usability comments) and one out-of-scope code (U – irrelevant and unknown) were defined, each code correspond to one intent in the chatbot. Each in-scope intent was provided with approx. 20 sample utterances. The sample utterances were not subsets of the protocols, they were generated independent of the actual protocols based on code definition and context knowledge (i.e., the expected verbalizations under the theme and their variations). A complete list of sample utterances is provided in Appendix D. Protocols outside the coding scheme were classified as "out of scope" (the U code) using a fallback intent (Amazon Web Services, 2020a). When the chatbot receives an input that it cannot recognize (e.g. protocols irrelevant to the defined coding scheme that it has not learnt before), the fallback intent is activated and a pre-defined response is given as the output. The fallback intent does not have sample utterances.

Table 5-4 Coding scheme

Code / Intent	Explanation	Example
I – app interaction	Protocols describing the interactions with the app	"click this button"; "now I'm looking at the map"
M – movement	Protocols about movement and navigation.	"I'm going to..."; "I'm crossing the street."; "I'll turn left here"
T – task-related	Protocols about the content of the fieldwork (.e., whether a building is a remnant)	"this building belongs to the old factory"
Y – usability comments	Protocols about the app's usability issues, including comments and recommendations	"the labels keep moving and it makes me dizzy"; "It would be better if you provide more information here"
U – out of scope (irrelevant and unknown)	Fallback code for protocols that the chatbot fails to recognize, including protocols that don't belong to the I/M/T/Y codes, and	N/A

To discover the usability issues, all protocols with code Y were extracted and usability issues were then manually identified and categorized based on their themes. Usability issues were reported in the most commonly viewed screens: the map, the AR and the take-note screen; additional information on POIs

were requested to better understand the past of the area; some recommendations were made regarding functionalities of the app. The following list summarizes the identified usability issues. A detailed result table with supporting protocols is provided in Appendix E.

Table 5-5 Usability issues discovered with the chatbot and manually grouped into themes

Theme	Issues raised by test persons
AR	- AR labels were unstable and should be attached (anchored) better. - AR was showing labels for POIs that were not visible (obscured by other objects).
Compass and map	- Compass failed multiple times. - The test person wanted a planimetric map and didn't like that it kept turning into oblique; satellite image was suggested as an alternative base map. - A traffic layer was suggested. - The test person suggested to show buildings as footprints instead of points.
Note-taking	- The test person didn't like the default title of "my note" because he had to delete it every time - The test person suggested to have a "save" button for the notes, instead of using the "back" button to save the note
Additional information for POIs	- More contextual information on the POIs was needed to understand the history. - The old maps should be annotated to highlight the remnants. - Some photos in the info page were taken from an angle that made the subject hard to recognize from other angles.
Restriction and safety	- The test person wanted information on which areas were open to the public and which were not. - The fieldwork led to an area without a sidewalk and the test person raised safety concerns.
Recommendations on functionalities	- Recommended ordering of the POIs; marking of the POIs that have visited - Directions to POIs in AR.

5.5. Integration: exploring mapped fixations with think-aloud protocols

Think-aloud protocols were also used to find possible explanations for patterns and interesting spots found in fixation data. After synchronization, the protocols were plotted along the same timeline with the fixations and can be interactive explored (e.g. with interactive plotting libraries such as Plotly; Plotly Technologies Inc., 2015). An example of using the timeline for interactive exploration is shown in Figure 5-11. Apart from the fixation sequence, the timeline for protocol segments is added at the bottom (gray), and the content of the protocol segment is shown on mouse hover. In this example, the test person verbalized about a building being a formal textile store before switching his fixation from the info screen to the building.

Figure 5-11 An example screenshot of interactive exploration of fixation sequence and think-aloud protocols. The test person verbalized about "this building might be the same as the textile store", while he read about the textile store described in the info screen of GeoFARA, and looked at the building at the same time.

Think-aloud protocols were used to discover possible explanations for the difference in switch frequency (Figure 5-9). Recording segments (1 minute in length each, corresponding to the minutes in Figure 5-9) with high overall switch count (≥20) and high phone-building switch count (≥5) were examined. It shows that high overall switch frequency is often related to movements: protocols can suggest the test person was walking to the next spot (e.g. "I'm first going to the villa Schuttersveld."). In those scene segments, interactions mainly happen between the map-AR screen and surroundings. Sometimes it is between the info screen and the surroundings when the person already knew the route to his next spot and already started to look at the info screen on the way there. And because of the walking, the fixations on the phone tend to be shorter, and there tend to be more switches between the phone and the non-building surroundings. At the same time, high phone-building switch frequency is often related to learning about the buildings and comparing the present and historical layout. The test person was often reading out the content on the info screen, comparing the building with the old map (e.g. "Sounds like this building actually was the same building as textile store"), or verbalizing about his understandings of the area's textile past (e.g. "Okay, so I see a cafe and bar, so that seems more like factory town type of locations"). It often involves longer fixations on the phone, especially on the info and old maps, which indicates a more careful reading of the information, instead of glancing while walking.

The protocols also give insight into the possible reasons for the change in the use of GeoFARA. For example, before switching from the map-AR to full map, the test person mentioned "… I don't really get a lot of value from the augmented part, so I think I'm just gonna close it." And when he returned to the map-AR screen later, he mentioned: "so this is an example where the augmented was helpful because I didn't have enough base map context." Locating these protocols to find explanations for discoveries found in fixation patterns is much easier when the protocols are synchronized and explored together with the fixations.

5.6. Integration: exploring mapped fixations and think-aloud protocols with location data

After synchronizing the timestamp of the fixations and the simulated GPS recordings, each GPS point could correspond to a list of mapped fixations. The most frequent fixation target in the list was assigned to the GPS point. The four scenes (*east, store, corner, middle*) with simulated routes and mapped fixations are shown in Figure 5-12.

The accuracy of the GPS measurements was not consistent during the simulation, especially under the tunnel, around trees, or next to big buildings. But the accuracy is considered sufficient for qualitative interpretation with the help of context knowledge.

Figure 5-12 shows the spatial distribution of the fixations along the routes. Fixations on the info and old maps often took place close to the POIs, note-taking often took place after a POI had been explored. It indicates that the learning took place when the POI was in sight and the information on the app could be directly associated with the POI in reality. The relationship between fixation and locomotion can also be preliminarily explored with the clusters of the points. The test persons mostly used the map-AR while walking, and stopped to take notes; the info screen were viewed both when walking and standing. However, because the pace of walking in the simulation was not exactly the same as the pace of walking during the actual fieldwork session, more detailed analysis on locomotion could not be done with this data.

The map shows that the environmental context might have an impact on the fixation patterns. The test persons had more fixations on buildings when they were surrounded by POI buildings (e.g. in the store area) than when the surrounding was less built-up (the area around KPN and Stichting 55+). When the environment context was the same, the difference in fixation patterns can provide insights on the tasks the test person was performing. This can be examined when the test person walked the same road more than once: the fixation patterns along that road (to and back from a POI) can be different because the test person was performing different tasks. In scene *east*, the test person had more interactions between the app and building when he walked towards the Menzis building and the tunnel (actively learning about the POIs); but barely looked at the app when he passed the same building on his way back after finished the learning. Similar pattern can be found in the *corner* scene: on his the way to Volkspark Tunnel, the test person looked at the info screen more, but on the way back mainly he paid more attention to the map-AR screen and the surroundings. Based on the recording, he was reading about the tunnel while walking towards it, since he already knew the location of the tunnel and didn't need navigation, and was not focusing on the fieldwork task but commenting on the AR functionalities on his way back. However, also in the *corner* scene, the fixation patterns are similar on the route to and back from the Stichting building: the test person looked mostly on the map-AR screen and the environment both ways. Based on the recording video, on the way to the Stichting building, he was relying on the AR to navigate to the building (because "the map was unclear"), and he only started to read the info screen when he was almost directly in front of the Stitching building. This case also shows the usefulness of the AR of helping the test person to associate the information shown in the app with the POI in reality.

It should be noted that, because the GPS points were assigned with the most frequent fixation target in the time span of the GPS measurement (1s), finer fixation patterns (e.g. quick switch of fixations) are not visible in the maps. Also, the sequence of the fixations becomes more difficult to find when the test person slows down or stops walking. The maps cannot replace the fixation sequence and they should be explored together in order to have a better understanding of the visual and physical behaviors.

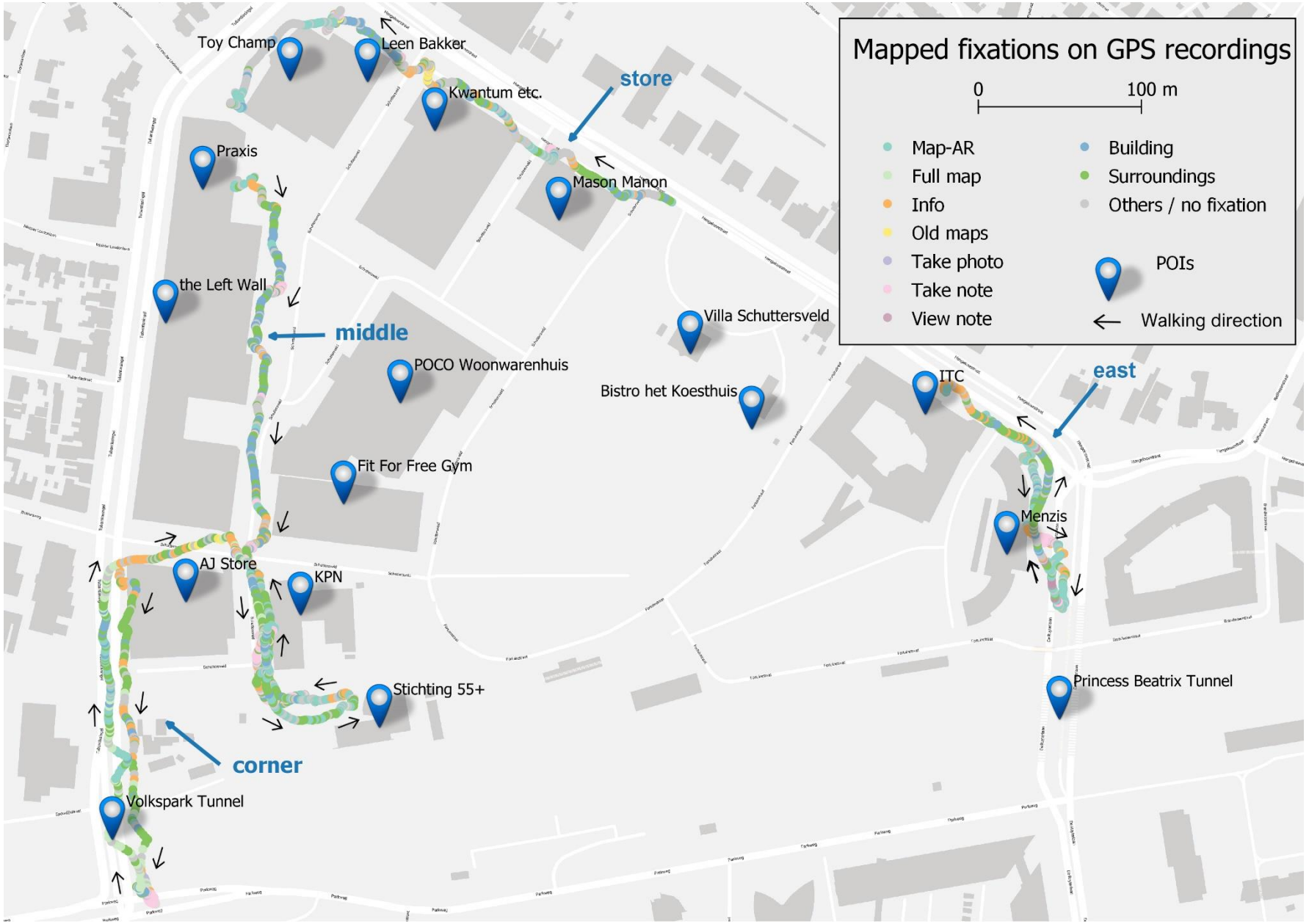


Figure 5-12 Mapped fixations and simulated GPS recordings after synchronization

The protocols can be explored in a similar manner with interactive plots. Figure 5-13 shows an example screenshot of interactive exploration of mapped fixations, think-aloud protocols and GPS recordings. The same exploration was conducted with the four scenes with GPS simulations. It shows that the test persons often verbalize about the fieldwork task (e.g. if a building belonged to the old factories) when they were close the POIs. This corresponds to the fixation patterns and indicates that the learning took place when they saw the POI in reality and associated it with the information provided by the app. Demands of more or clearer information on the POIs were often verbalized during the learning process close to the POIs. Recommendations on app functionalities were often verbalized during the walk between POIs, when the test person was less occupied with actual fieldwork activities.

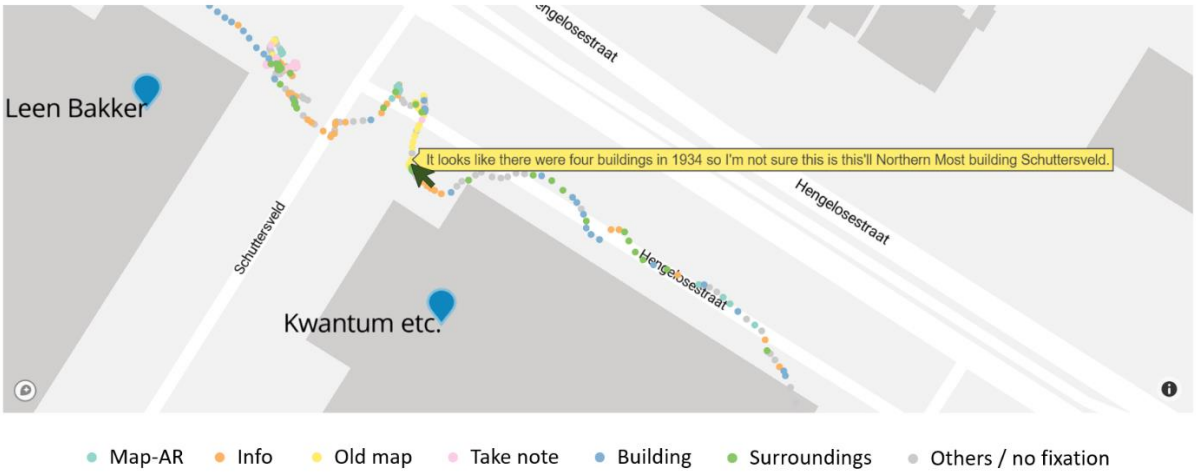


Figure 5-13 An example screenshot of interactive exploration of mapped fixations, think-aloud protocols and GPS recordings. The test person was looking at the old map in front of the Kwantum building, and verbalizing about if this building was the most northern one on the old map

5.7. Summary and mini-conclusion for the GeoFARA case study

This chapter demonstrated the functionalities of the protype solution with the GeoFARA case study as a proof-of-concept. Simulations were conducted for the screen-recordings and location data that had not been collected on-site during the original experiment. Fixations were mapped to four object categories: cell phone, building, surroundings, others; and seven screen-content categories (screens). The distribution and sequence of the fixations were explored. Think-aloud protocols were used to identify usability issues, and were explored together with eye-tracking data to support and explain the patterns and findings of the visual behavior. The location data was explored together with mapped fixations and think-aloud protocols, it provided spatial and environmental context to the exploration of visual attention and verbalizations.

The demonstration analysis of the six scenes resulted in the following findings regarding the use and usability of GeoFARA:

- *The use of GeoFARA and the learning during fieldwork:* the most frequently viewed screens were map-AR (for navigation), info, and note-taking. Both the map and the AR were used for navigation. But the two test persons were depending more on the map to discover the remnants, the AR was used more when the map couldn't provide enough navigational information. When learning about the history of the area, the test persons relied on the info screen. Old maps were also studied to learn about the historical layout. They discovered and decided on the next destination (POI) with the map, used the map or the AR to navigate to it, read and took notes about it when they reached it. Reading the info often happened while the test persons were standing close to the POI, sometimes while walking towards the POI; looking at the old maps mainly happened when

the test person stood close to the POI. GeoFARA helped guide the test persons to discovered the remnants, and helped to engage the buildings in reality with the remnants from the historical industry.

- *Usability issues of GeoFARA:* The AR labels should be more stably anchored, and should only appear when the actual POI is visible. More information on the POIs is needed to provide more context, and the photos attached in the POI info should be taken from an angle that allows the user to quickly connect the photo with the POI in reality. More annotations on the old maps are needed, for example highlighting the currently visible remnants, to help the user relate the current building in reality with the building on the old maps.

The next chapter will carry out a preliminary evaluation on the performance of the prototype in this case study. The limitations and the possibilities for more generic use of the prototype will also be discussed.

6. PRELIMINARY EVALUATION AND DISCUSSION: GEOFARA AND BEYOND

6.1. Introduction

This chapter presents an evaluation and discussion about the prototype solution. A preliminary technical evaluation is performed with the data from the GeoFARA case study (Section 6.2). The automated fixation mapping results are compared with the results of manual mapping; the protocols coded with the chatbot are compared with manual encoding; the execution time during the case study is presented as a general indication of the time performance of the prototype. The discussion (Section 6.3) addresses the information that can be derived with the prototype, the limitation in the prototype and possibilities for future work. The discussion has its basis on the case study, but it also aims at a wider context of using mobile eye-tracking in GI user research.

6.2. Preliminary technical evaluation with case study data

6.2.1. Mapping fixations to real-world objects and screen contents

The fixation mapping results from panoptic segmentation were compared with manual mapping. Because of the manual workload, not all scenes were manually mapped. Three scenes (*villa, store, wall*) were selected for the comparison. The *villa* and the *store* scenes were selected because they involve some most typical POIs (the Villa, and store buildings on old factory sites). The *wall* scene was selected because it includes a unique scenario where the POI *Left Wall* was partly covered with ivy, which could introduce errors in fixation mapping. A total of 3120 fixations were involved in the comparison. For this comparison, and the other comparisons later in this chapter, the purpose was to give a preliminary indication of the performance of the prototype in the scenario of the case study instead of precisely measuring the accuracy and efficiency, so only one human operator was involved for the manual operations.

For fixation mapping to real-world objects, the comparison was based on the four object categories. The consistency between fixation mapping with panoptic segmentation and manual mapping in the three selected scenes is shown as confusion matrices in Table 6-1, 6-2, and 6-3. In the manual mapping, the "others" category was not involved because all fixations were semantically mapped to cell phone, building, or surroundings.

Table 6-1 Confusion matrix: manual and automated fixation mapping to real-world objects, scene *villa*

auto / **manual**	building	cell phone	surroundings	others	total	**recall**		
building	44	0	27	3	74	**0.59**		
cell phone	1	427	11	1	440	**0.97**		
surroundings	7	2	326	6	341	**0.96**		
others	0	0	0	0	0	**n/a**		
total	52	429	364	10	855			
precision	**0.85**	**1.00**	**0.90**	**0**			**consistency**	**0.93**

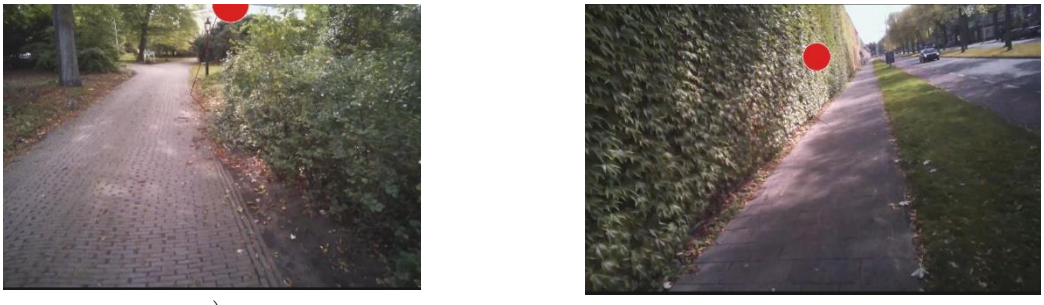
Table 6-2 Confusion matrix: manual and automated fixation mapping to real-world objects, scene *store*

auto / manual	building	cell phone	surroundings	others	total	recall		
building	167	0	23	11	201	**0.83**		
cell phone	1	790	7	2	800	**0.99**		
surroundings	24	5	201	20	250	**0.80**		
others	0	0	0	0	0	**n/a**		
total	192	795	231	33	1251			
precision	**0.87**	**0.99**	**0.87**	**0**			**consistancy**	**0.93**

Table 6-3 Confusion matrix: manual and automated fixation mapping to real-world objects, scene *wall*

auto / manual	building	cell phone	surroundings	others	total	recall		
building	206	0	49	11	266	**0.77**		
cell phone	2	503	2	0	507	**0.99**		
surroundings	16	1	222	2	241	**0.92**		
others	0	0	0	0	0	**n/a**		
total	224	504	273	13	1014			
precision	**0.92**	**1.00**	**0.81**	**0**			**consistency**	**0.92**

In the three selected scenes, the automated approach is relatively consistent with the manual approach with overall consistency above 0.9. The cell phone is very well recognized by the automated approach in all three scenes, with both recall and precision above 0.97. Part of the inconsistency here might be caused by the manual mapping error of the fixations landing on the edge of the phone. There is some confusion between building and surroundings. By reviewing the video and comparing the result, it is found that inconsistency often happens when the building is far, and/or when there are obstructions (e.g. trees) in front of the building. The confusion between the building and surroundings categories in the *villa* scene (0.59 recall for building) mainly happens when the building is far away and less identifiable without context knowledge (Figure 6-1a). The inconsistency between building and surroundings in the *wall* scene (0.77 recall for building) is mainly because fixations landing on the ivy on the wall were mapped to "tree" and further grouped to the surroundings category (Figure 6-1b).



a) b)

Figure 6-1 Examples of misclassified fixations (red circle in the images). a) fixation on far, blurry building; b) fixations on ivy on the wall

The screen contents identified by the automated approach were also compared with manual fixation mapping. The confusion matrices from the same three scenes are shown in Table 6-4, 6-5 and 6-6. The results show that screen contents are generally distinguishable by the automated approach in the three scenes (consistency scores of 0.71, 0.89, and 0.83 respectively). The info screen is relatively well-identified by the automated approach in all three scenes, the old-map screen is well identified in the *store* and *wall* scenes. The take-note screen scores high in recall in all three scenes (over 0.9), but has a low precision

score in the villa section (0.55). This might be caused by a small human error during the simulation of the *villa* scene, where the take-note page was mistakenly opened.

Confusion can happen between similar-looking screen contents, for example between the map-AR and take-photo screen. Because the AR part of the map-AR screen is the live feed from the camera, part of the features vector (HSV histograms) of the map-AR screen might be similar to that of the take-photo screen. And when the simulation was conducted indoor, the scene from the camera was relatively similar throughout the simulation. It could make these two screens appear more similar than they would be during the actual fieldwork.

Table 6-4 Confusion matrix: manual and automated fixation mapping to screen contents, scene *villa*

auto / manual	map-ar	full map	info	old map	take note	take photo	total	recall		
map-ar	155	0	6	5	28	24	218	**0.71**		
full map	3	0	0	0	0	2	5	**0.00**		
info	6	0	73	7	3	0	89	**0.82**		
old map	0	0	5	13	2	0	20	**0.65**		
take note	3	0	0	0	52	3	58	**0.90**		
take photo	16	0	0	0	10	11	37	**0.30**		
total	183	0	84	25	95	40	427			
precision	**0.85**	**n/a**	**0.87**	**0.52**	**0.55**	**0.28**			**consistency**	**0.71**

Table 6-5 Confusion matrix: manual and automated fixation mapping to screen contents, scene *store*

auto / manual	map-ar	info	old map	take note	total	recall		
map-ar	221	13	10	17	261	**0.85**		
info	19	189	1	8	217	**0.87**		
old map	1	5	109	6	121	**0.90**		
take note	3	0	0	188	191	**0.98**		
total	244	207	120	219	790			
precision	**0.91**	**0.91**	**0.91**	**0.86**			**consistency**	**0.89**

Table 6-6 Confusion matrix: manual and automated fixation mapping to screen contents, scene *wall*

auto / manual	map-ar	full map	info	old map	take note	take photo	total	recall		
map-ar	53	2	11	3	1	8	78	**0.68**		
full map	0	74	7	2	12	0	95	**0.78**		
info	7	9	141	1	3	0	161	**0.88**		
old map	0	0	1	39	4	0	44	**0.89**		
take note	0	1	0	0	92	0	93	**0.99**		
take photo	11	1	0	0	2	18	32	**0.56**		
total	71	87	160	45	114	26	503			
precision	**0.75**	**0.85**	**0.88**	**0.87**	**0.81**	**0.69**			**consistency**	**0.83**

Human errors were introduced during the simulation, and would later result in the inconsistency between the manual and automated approach. The manual and automated mapping results were compared on a timeline and errors and delays in the simulation were be examined. Figure 6-2 shows an example of the timeline for a section in scene *store*. The green dots are fixations that were mapped consistently by the manual and automated approach; when the results are not consistent, blue dots are the result from automated mapping and orange dots are the result of manual mapping. There were many actions involved

in a 15-minute long simulation session, where mistakes could happen. As a result, not every action shown in the eye-tracking video recording had been correctly captured in the simulation, especially when the test person performed a sequence of actions in a relatively short time (e.g. open a page then quickly close it). The delay in simulation actions can cause inconsistency at the beginning or end of periods of continuous attention. But these errors would not be a problem when the screen-recording video is recorded on-site, simultaneously with the eye-tracking data.

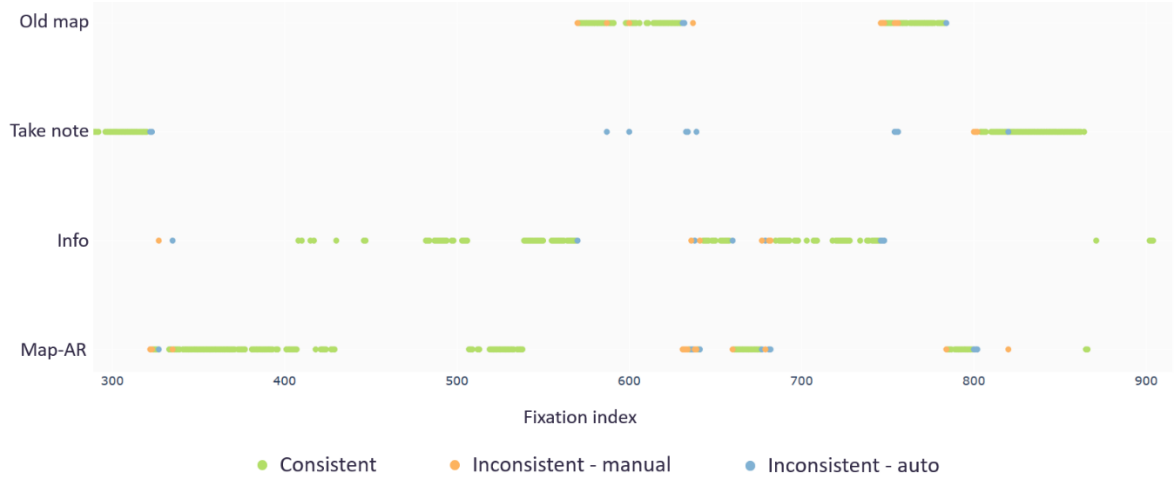
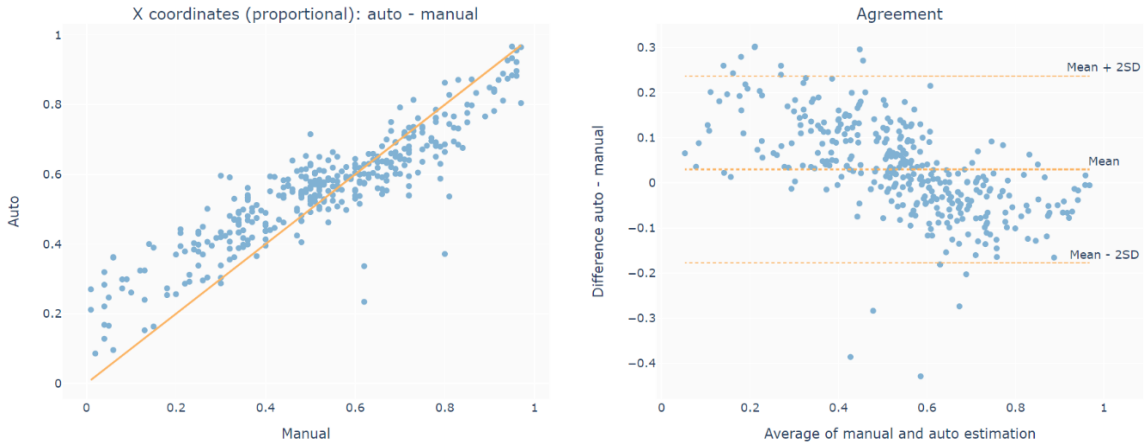


Figure 6-2 An example of visualizing the inconsistency between manual and automated fixation mapping to screen-contents, from scene *store*

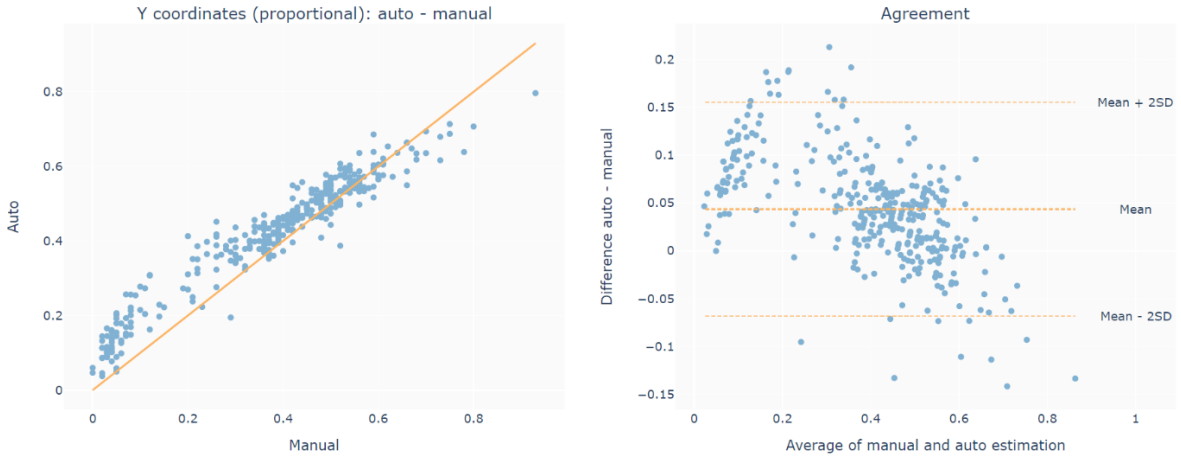
6.2.2. Mapping fixations to screen-coordinates of the mobile display

The assumptions on the visibility and position of the phone (Section 4.3.2.2) were examined by manually inspecting the recordings. The recording shows that the phone was held relatively upright and the entire phone was in the camera view most of the time. When the phone was held very close to the person, part of the phone might go out of the camera view, but it was not common in the recordings.

The estimated screen-coordinates were compared to the manually-mapped results. For the *villa* and *wall* scenes, fixations on the map-AR screen were manually mapped to the corresponding locations on a reference image. A total of 359 fixations were involved in the comparison. The result is shown in Figure 6-3. The X and Y coordinates are not absolute coordinates, but relative coordinates in proportion to the width and length of the screen. They are compared separately because the phone is rectangular, and the same amount of error in relative proportion will be different in screen-coordinates expressed in pixels. It should be noted that manually mapping fixation to exact screen coordinates is not very precise either, especially when the reference image could not always be exactly the same with the screen content in the recording due to dynamic user interactions, the results are only used to indicate the agreement between the two methods, they are not meant to evaluate the accuracy of either manual mapping or the automated estimation.



a) proportional X coordinates



b) proportional Y coordinates

Figure 6-3 Comparison of estimated and manually mapped (proportional) screen-coordinates. X and Y coordinates are compared separately because the phone is rectangular, and the same amount of error in relative proportion will be different in screen-coordinates expressed in pixels

The automated estimation of fixation screen-coordinates generally agrees with manual mapping (X: mean difference=0.03, SD=0.10; Y mean difference=0.04, SD=0.06). The automated estimation tends to agree with manual mapping around the center of the screen and deviates more towards the edges. The estimated coordinates tend to drift towards the center of the screen. The (0, 0) origin is in the upper left corner. The estimated X and Y coordinates tend to be larger than the manually mapped towards the left and upper edges, and smaller than the manually mapped towards the right and lower edges of the screen. This could be caused by the non-upright position of the phone (Section 4.3.2.2 and Figure 4-5). A tilted phone will result in a non-rectangular mask and distort the minimum rectangular bounding box. The estimated coordinates of the fixation based on a distorted bounding box may drift towards the center, while the actual direction of the drift depends on the location of the fixation. The distortion and drift will increase with the tilt angle of the phone. The generally agreeing result being automated and manual mapping also suggest that the drift is not drastic, which in turn indicates that the assumption of no drastic tilt of the phone was generally satisfied. It was also discovered that sometimes the protective cover of the phone disturbed the panoptic segmentation and resulted in a distorted phone mask (Figure 6-4).

Figure 6-4 An example when the protective cover of the phone caused distortion of the instance mask (the mask was larger than it should be when part of the protective cover was also segmented as "cell phone"). Note: semantic masks are not plotted in this figure.

The agreement between the automated and manual approach shows that the automated estimation of fixation screen-coordinates is reasonable enough to be used to indicate which part of the screen (e.g. upper left, lower right) is being attended. However, because of the accuracy of the eye-tracker and potential errors introduced during calibration, the screen-coordinates of fixations (both manually mapped and automatedly estimated) might not be accurate enough to identify the exact screen feature (e.g. map symbols) on the mobile display.

6.2.3. Coding think-aloud protocols

The encoding by the chatbot was compared with manual coding. Because the chatbot encoding is context-independent, the comparison with manual coding will not distinguish between the scenes. Manual coding was also performed context-independently, where the human coded protocols in a random order to minimize the influence of contextual information. A total of 333 protocols were involved in the comparison (from scene *villa*, *store* and *wall*). The result is shown in Table 6-7.

Table 6-7 Confusion matrix: manual and automated coding of protocols

auto / **manual**	I	M	T	Y	U	total	**recall**	
I	16	0	3	3	4	26	**0.62**	
M	6	21	2	5	3	37	**0.57**	
T	3	3	63	17	11	97	**0.65**	
Y	5	1	7	45	4	62	**0.73**	
U	11	5	13	22	60	111	**0.54**	
total	41	30	88	92	82	333		
precision	**0.39**	**0.70**	**0.72**	**0.49**	**0.73**		**consistency**	**0.62**

The encoding of the chatbot is not very consistent with manual encoding (overall 0.62 consistency with 5 classes). But given the relatively low volume of sample utterances used to train the chatbot, the result is reasonably acceptable. The major reason for the low consistency is that the chatbot had trouble predicting out-of-scope data (the U class, with 0.54 recall). It tried to allocate a code even when the input sentence was irrelevant to the defined coding scheme. It performed reasonably well with the in-scope predictions (consistency of the four in-scope class [I, M, T, Y] was 0.73). Meanwhile, out-of-scope prediction (identifying the "unknown" class) is a challenging problem in general: while state-of-art models such as fine-tuned BERT can achieve high accuracy on in-scope predictions, their performance can be significantly lower (approx. 0.5 recall) on out-of-scope predictions (Larson et al., 2019). Among the in-

scope code classes, M (movement) and T (task-related) score relatively higher precision (0.70, 0.72 respectively); Y (usability comments) has higher recall (0.73). It indicates that approx. 3/4 of the usability issues in the verbalization were successfully identified by the chatbot. An inspection of the result shows that the chatbot was also able to make inferences over protocols that are not explicitly stated in the sample utterances. For example, the sample utterances for the T class focus on general expressions on whether a building is a textile remnant, but don't contain any specific POI names, but the chatbot was able to correctly classify many specific POI-related sentences as task-related (e.g. "So I wonder, is Schuttersveld the name of the person within this [Jan] van Heek purchased it…" and "Okay, so this was, uh, barren's house, and it seemed as though there were two different people that owned this").

The sample utterances could have an influence on the encoding result. It is found that the chatbot performs relatively well with shorter simple sentences, and sometimes has difficulties with longer, compound sentences (e.g. sentences that make a statement then explain reasons). This might be a result of the sample utterances only having shorter sentences. The complex sentences were not included in the sample utterance because the original idea was that sample utterance would try to capture characteristics in the language that are directly linked with the intent, and compound sentences that include reasoning/explanation contain information not directly linked with the intent might confuse the chatbot. However, since compound sentences are a part of natural speaking, including examples for complex sentences on top of the simpler ones might help to improve the encoding result by making the training samples more similar to the actual protocols to be coded.

On top of the sample utterances, the quality of transcription and segmentation of the protocols can have a direct impact on the encoding. Errors in transcription can leads to the misclassification of the protocols. Although sentences can be considered as a suitable unit for segmentation, it might not be suitable for all the protocols, as some sentences may contain multiple meaning-units. For example, consider the following protocol directly extracted from the transcription result: "That's kind of unclear from the description, but whereas these other buildings look like they're older, look like they're newer, this one possible." The two clauses in the sentences have different intentions (usability comments: pointing out unclear information, and task-related: making inferences about the buildings), but only one code will be allocated because they belong to the same sentence.

6.2.4. Execution time

The actual execution time of the prototype solution depends on the hardware configuration. Table 6-8 gives an indication of the time performance under the hardware configuration during the case study. The case study was performed on a Windows 10 laptop computer with NVIDIA Quadro P1000 GPU and Intel i7-8750H (2.20GHz) CPU, 16GB RAM. The time performance of fixation mapping and screen-recording processing mainly relies on GPU capabilities; the processing of think-aloud audios is performed remotely with AWS and is less influenced by the local hardware configuration.

Table 6-8 Indications for execution time

Processing element	Time indication
Fixation mapping to real-world objects	0.7 second / fixation
Fixation mapping to screen contents	1 second / fixation
Encoding protocols	0.5 second per protocol

The speed of mapping fixations to real-world object is influenced by the resolution of the input image video frame. In the prototype solution, the frame is not resampled. If needed, the speed of fixation

mapping can be further improved by resampling the frame image to a lower resolution. The prototype makes use of the state-of-art panoptic segmentation model, but very recently, a novel single-shot panoptic segmentation model is proposed that is claimed to be able to perform at 21.8 frames per second (FPS) (Weber, Luiten, & Leibe, 2019). With the fast progress in computer vision and segmentation models, making use of such novel models will dramatically enhance the time performance of mapping fixations to real-world objects. Mapping fixations to screen contents take two steps: indexing the candidate images and searching for a best-match (Section 4.3.3). The indexing phase will take extra time, but one set of candidate images only needs to be indexed once. The speed of indexing and searching depends heavily on the number of bins used to calculate the histograms and the size of the images, the indication given in the table is based on the HSV bin configuration of (20, 20, 30).

6.3. Discussion

6.3.1. Deriving information with the prototype solution

6.3.1.1. Linking the reality, the representation of the reality, and the mental map

The proof-of-concept case study shows that the requirements discussed in Section 4.2 have been addressed: fixations can be automatically mapped to real-world objects and screen contents on the mobile display; think-aloud audios can be processed with a semi-automated workflow, and the result can be linked with eye-tracking data; the location data can be linked with the eye-tracking and think-aloud data and analyzed together.

The case study demonstrates that the prototype enables information about visual attention to be derived without laborious manual annotations. By mapping fixations to real-world objects and screen-contents, the fixations are directly associated with object semantics without using a reference image. The distribution and the sequence of visual attention can be extracted more easily with fixation-based metrics. The distribution metrics (e.g. counts and durations) indicate the amount of attention spent on the display and the environment (e.g. the excessive amount of attention on the phone in the GeoFARA case); they can highlight the salient or interesting objects in the environment, as well as the frequently used screens on the mobile application, and suggest the cognitive function level during information processing. The sequence metrics (switches, revisits, etc.) show the change of attention between the reality and the representation (over time). They can indicate the processes such as search and learning. In the GeoFARA case, higher switches frequencies between buildings and phone might be associated with the learning process. With the prototype, a timeline can be easily produced with the mapped fixations and the sequence can be easily explored, which provide insights into the search and learning strategy, and can also suggest the change of the strategies over time.

The think-aloud protocols provide immediate insights into the mental process as the test person interacts with the reality and its representation. The chatbot was able to perform a classification task on think-aloud protocols with reasonable accuracy, especially considering the dramatically lower volume of training (sample utterances) compared to conventional natural language classification solutions. It successfully identified a large part of the usability issues, and also showed the potential to generalize over the given sample utterances and make predictions out of the given samples. It shows a possibility to process the protocols and derive information with semi-automated assistance instead of going through the entire transcript manually. After the transcription-segmentation-encoding pipeline, the protocols can be used to identify usability issues experienced or comments made by the test persons, which might not be directly visible from the eye-tracking data. The verbalizations also provide insights on the mental map of the test person, and how that mental map is "updated" as the test person processes the information from the

mobile display and the environment (e.g. discovering a new remnant of the textile industry). Combining the protocols with eye-tracking data and exploring them on one timeline, the patterns and discoveries in the fixation can be supported and explained by the verbalizations.

The location data adds context to the visual behaviors and verbalizations. In addition to *what* is looked at and verbalized about, it provides information on *where* the test person is looking and verbalizing from. In the case study, the *where* part can directly indicate how the test person used/learnt with GeoFARA. Reading the info screen in front of the POI and reading the info screen on the way to it suggest two different kinds of learning: whether the learning about the POI took place when the POI was in sight (i.e., whether there were active visual interactions between the POI described on GeoFARA and the actual object in reality). As the test person solves a spatial problem in the real-world environment, the visual and physical behaviors and the verbalizations (mental processes) are influenced by both the task and the environment. For example in the GeoFARA case, more attention on buildings in some areas could be related to the area being more built-up, and not necessarily because the test person was intentionally more focused on the buildings (e.g. the *store* scene versus the *villa* scene). The location data adds the spatial and environmental context into the analysis of fixations and verbalizations, and can assist the exploration of this environmental influence. It also combines locomotion into the analysis, which also provides intentions, and might be included in sequence-based analysis of the visual attention (Kiefer et al., 2011).

6.3.1.2. Beyond GeoFARA: some possibilities for other types of analysis

The case study mainly demonstrates the prototype in qualitative exploratory analysis for a less constrained, exploratory experiment. But it can also support quantitative/statistical analysis on more constrained, quantitative experiments. With exploratory analysis, the main role of the prototype is to assist the discovery of potential patterns and unique spots in the data, and reduce the amount of manual inspection of the recording video. Instead of inspecting the entire video to find commonalities and unusual behaviors, the summarized statistics plots can already give hints on the general patterns and potential outliers, and the recording videos can be reviewed with focus. The exploration of synchronized fixation-verbalization-location data can provide the overview (of a single test person or an entire group of test persons), as well as the details of the interaction between the environment, the representation and the mental process. For quantitative analysis, the data generated by the prototype can be directly used for statistical purposes. And because of the automation, the manual workload would be less constraining on the number of test persons recruited in an experiment, which provides more opportunities to apply quantitative methods. The processed data can be analyzed with other analytical software or in GIS. The case study shows an example of performing independent analysis on a few scenes. But the prototype can also support analysis between groups in group experiments. Apart from analyzing the data per participants or recording scenes, the data generated by the prototype can also be merged and queried based on fixation targets (e.g. querying dwell time of a certain object to get an overview of how the object is inspected by test persons of different groups). It will facilitate the comparison between groups (e.g. groups with different interface designs or navigation assistants, for example, Schnitzler et al., 2016).

6.3.2. Limitations and possibilities

6.3.2.1. Fixation mapping: real-world objects, screen contents, and screen coordinates

The evaluation shows the result of the automated fixation mapping (to object categories) has good consistency with manual mapping. However, the case study only represents one scenario in an outdoor urban environment with a relatively simple object categorization (cell phone, building, surroundings). The performance and the suitability of the proposed method still need to be tested in other scenarios with

other categorization schemes. In particular, the indoor scenario might be potentially more complicated. Segmentation models tend to have lower performances on indoor objects (door, stairs, etc.) than outdoor "stuff" classes (sky, road, building, etc.) (Zhou et al., 2019). And when mapping fixations to object categories, the overall accuracy of the fixation mapping does not only depend on the accuracy of the segmentation model on individual object classes, the categorization of the objects will also have an impact. Indoor landmarks are often grouped into more categories (for example, Viaene et al., [2016] defined 19 categories for indoor landmarks), which will potentially bring more challenges to accurate fixation mapping.

One observation from the case study is that the semantic information obtained from the panoptic segmentation model (after categorizing to the four object categories) is not always consistent with the semantic information obtained with manual fixation mapping. For example, in the case study, a fixation was mapped to "paper", and was then grouped to the "others" category. The recording shows that the fixation was indeed on a piece of paper lying on the ground. However, with manual mapping, this fixation would be considered as a part of "ground", in the "surroundings" category. A similar situation happened in the *wall* scene, where the fixations landing on the ivy on the wall were mapped to "tree" and further grouped to the "surroundings" category; with manual mapping, because the ivy was on the wall, these fixations should be semantically mapped to the wall (building category). Although these cases are not very common the case study, it implies a discrepancy in fixation semantics between the object-based automated (segmentation) method and manual mapping: when the fixation is automatically mapped correctly to the object, it might not be mapped with the same semantics as compared with manual mapping.

In the case study, one major problem with the object-based fixation mapping is that individual buildings cannot be distinguished (i.e., the semantic information is on the "building class" level), and it is not certain whether the building the test person was interacting with was a POI (remnant of the textile industry). In this particular case study, with GPS recordings, context and background information, sometimes the individual building being viewed can be inferred. But in a different scenario, for example, a navigation experiment in a built-up area, only knowing the fixation is on "a building object" might not be sufficient. For example, in the study of Kiefer et al. (2014), individual buildings were considered as different AOIs. In some cases, such as in Hollander et al. (2019), the AOI is not an entire building, but a part of the building (e.g. a potentially salient part of the building). In these cases, when the "building category" needs to be broken down into "identifiable building instances" and 'identifiable parts of the building", more semantic information is needed for the fixations. However, this information might not necessarily come directly from image segmentation models, because in common practice, buildings are still amorphous areas that are not suitable for instance segmentation. For example, Ogawa & Aizawa (2019) proposed a method that builds upon the results from image segmentation models to delineate building instances on street view images with the help of a map and the location and camera position information attached to the street view image. It is an example where the semantic information obtained from image segmentation models can be enhanced with other data, and provides an opportunity for enhancing semantics in fixation mapping, especially when the location data can be recorded in outdoor research.

During manual mapping, the human decision on fixation semantics is based not only on *what object it is* but also *where the object is* in the scene and with respect to other objects: the ivy on the wall is part of the wall; the building on one side of the street is different from the building on the other side. The automated mapping solutions based on image and feature-matching algorithms (e.g. the Tobii RWM) focus more on the *where* part of the semantics by first transforming image coordinates from video frames to the reference image, then attaching semantic information to the reference image by defining AOIs on it. The object-based mapping solution focuses on the *what* part of the semantics by performing image segmentation and

object detection directly on the video frames. An integration of object-based and feature-matching methods (e.g. matching segmentation results with street view images) might be considered in the future to map fixations with richer semantics that are more consistent with human scene-understanding.

Regarding the screen-content on the mobile display, the methods used in the prototype are demonstrated to be suitable for multi-screen applications such as GeoFARA, but may not be as suitable when the screen only shows one interactive map (for example in the study of Brügger et al., 2019). It will be more difficult to define discrete candidate images on such continuous interactive maps, and the differences between the candidate image might be relatively small for the algorithm to distinguish them effectively. Its performance with other kinds of screen contents still need to be further investigated. And image descriptors such as shape-matching-based descriptors could also be considered here.

It should be noted that there is a difference between the screen content a fixation is mapped to and what the test person is actually looking at. The *screen content* is a representation that describes the screen as a whole, while the test person only looks at *one screen feature* at a time. To establish the link between the reality and the representation, sometimes more detailed information on what exactly is viewed on the screen (e.g. a particular map feature) is needed. In order to support this, the estimation of the fixation screen coordinates has to be further improved. The current estimation of fixation screen coordinates can indicate which part of the screen the fixation lands on (e.g. the upper left corner), but it is not accurate or precise enough to pinpoint the map features being viewed. The case study shows that using a mobile phone without a protective cover might help to improve the quality of the estimation. Having the test person hold the phone as upright as possible throughout the entire experiment will also likely improve the estimation, but it is not realistic and contradicts the goal of using mobile eye-tracking to capture natural behaviors in the real-world environment. Methods that help to orient the position and rotation of the cell phone might still be considered, including placing small markers on the corners of the phone (as done by Müller, Buschek, Huang, & Bulling, 2019). Combining user logging might also help identify the features being viewed on the mobile display. Similar to the methods of Göbel et al. (2019) and Ooms et al. (2015) on screen-based eye-tracking, when a web map is used on the mobile phone, if the user interactions with the mobile web map (e.g. pan, zoom) are logged detailed enough, these interactions can be reconstructed after the eye-tracking session, where the extent of the web map can be calculated, then the map feature being viewed can be queried given the estimated screen coordinates of the fixation. This approach will likely require good calibration of the eye-tracker, good estimations of fixation screen coordinates, as well as preparation (programming) on the mobile device to log user interactions intensively.

One overarching issue regarding the mapping is that the prototype works with filtered fixation data instead of raw gazes, as the same is usually practiced in manual mapping. But the configuration of the fixation filter might bias the mapping result (Göbel et al., 2019; Tobii Pro, 2019b). When the object is moving, the same gaze filter can yield different results when processing gazes in the coordinate systems of the eye-tracker (gazes in the scene camera video) and gazes in the coordinate system of the object (gazes mapped to snapshots). The effect of gaze filtering on fixations on reference images (snapshots) have been discussed (a more detailed description on the gaze filter issue can be found in Tobii Pro, 2019). But the influence of gaze filtering on fixations mapped with object-based mapping still needs investigation (e.g. how can gazes be aggregated based on their dispersion or velocity when they are mapped to an "object class" without coordinates information?). On the other hand, operating on raw gaze data will take a much

longer time because of the large volume of gaze data, but it yields "raw" results that allow researchers to compute fixations with their own parameter configurations.

6.3.2.2. Processing think-aloud protocols

Although the chatbot had a reasonable performance encoding the think-aloud protocols in the case study, they are not originally intended for this kind of use, and more testing with more protocols and different coding schemes are needed to determine whether they are robust enough to be used beyond the GeoFARA case study, in a wider context of think-aloud data collected in GI user research. Although state-of-art natural language processing models take much more effort to train and prepare, they can provide more robust performance when the amount of protocols to be encoded is large and the demand for encoding accuracy is high.

However, if a chatbot is used as an off-the-shelf protocol-encoder, improving the quality of the sample utterances can further improve the performance. The quality of sample utterances can have two aspects: how well they represent their corresponding intent, and how well can they represent the protocols to be encoded. The sample utterances can be generated based on theory and context knowledge (e.g. by analyzing the task and predicting what the test person may say), or by sampling the protocols and labelling them, or a mixture of both. Including the real utterances from the protocol might help the chatbot to learn the characteristics of them and improve the encoding result. Further, similarity metrics (e.g. cluster validation metrics such as cohesion and separation) can be used as a guideline to assess the quality of the sample utterances prior to building the chatbot, and the sample utterances can be improved if the sample utterances from different intents are too similar (Google Cloud, 2020). Nonetheless, because chatbots are offered as a black-box service which makes tuning and monitoring impossible, and the intention of using chatbots as a protocol encoder is to make use of an easily accessible solution and not to tune a "perfect" model, finding the "right" set of sample utterances will probably still remain a trial-and-error process.

The prototype performed context-free encoding with mutually exclusive code categories. It was considered as a simpler classification task (compared to context-aware or multi-class classification) for this prototype, as chatbots haven't been used for this kind of purposes before. However, depending on the experiment, the encoding process is not necessarily context-free, and the coding schemes might not be mutually exclusive (Yang, 2003). Such encoding will be more challenging to address with chatbots. Although the recent development in dialog systems is enabling chatbots to be more context-aware, such as by storing and modelling previous inputs (Aujogue & Aussem, 2019), modelling the context in chatbot services is not as straightforward as modelling the intents, since the "context" in think-aloud protocols cannot be directly modelled into a series of attributes (for context management in Lex, see Amazon Web Services, 2020d). Dealing with overlapping coding schemes is also currently challenging with chatbot services because generally intents should not overlap (i.e. a "utterance conflict" will be flagged when an sample utterance maps to two or more intents, Amazon Web Services, 2019).

6.3.2.3. Integrating data within the mixed-methods approach

The accuracy of the location measured by the mobile phone is acceptable for visual interpretation and exploratory analysis in the case study. But it still needs investigation whether the measurements are good enough for more quantitative analysis, especially in more built-up environments where tall buildings can impact the GPS signal. The case study only used the location-fixation data for visual interpretation and exploratory purposes. The main reason was that both GPS measurements and screen-recording videos were the result of simulation instead of on-site data collection. The errors from multiple sources introduced by the simulations propagated during the integration of mapped fixations and location data,

and the resulting error was hard to estimate. The propagation of the error needs to be quantified and evaluated first before further quantitative analysis can be conducted with the integrated data. In an ideal situation where the GPS and screen-recording video are acquired simultaneously with the eye-tracking data, the error sources will be more manageable and the propagation can be easier estimated.

With the GPS recordings, the location of the test persons is known, but the location of the fixation target is still not known. Knowing the location of the fixation target can be necessary when the targets are investigated based on the characteristics of their locations and their surroundings. For example, in the study of Wenczel et al. (2017), acquiring landmark structural salience requires information on whether the landmark is close to a corner and whether it is in or against the turning direction of the test person. In order to have this information, more data on head and body position is needed. Together with 3D city models (e.g. OSM buildings), the integration of location data, motion sensing, and eye-tracking data would enable the fixations to be mapped to individual buildings or other features. Although head-tracking is not currently supported in the mobile eye-trackers, alternative methods of acquiring the information still provide some possibilities. For example, the bearing of the mobile phone can be measured with built-in sensors, and solutions have been proposed with additional inertial measurement units (IMUs) mounted on the test person (Lander, Herbig, Löchtefeld, Wiehr, & Krüger, 2017; Tomasi, Pundlik, Bowers, Peli, & Luo, 2016). However, these methods are not yet commonly practiced in (geo) applications and their accuracy and suitability still need to be investigated (e.g. Lander et al. [2017] reported disturbance to the head-mounted IMU due to electromagnetic induction when the eye-tracker was running).

In the case study, data from the mixed-methods approach (eye-tracking, think-aloud, GPS recordings) was mainly used qualitatively for exploratory purposes. With more test persons and a more constrained experiment, the prototype can be tested with more quantitative analysis. Also considering the volume of the gaze data, data-driven and data-mining approaches might also be considered, for example, task inferences (Liao et al., 2019) and landmark inference (Lander et al., 2017).

6.3.2.4. Using the prototype solution

Because the first-stage prototype mainly aims at functionalities instead of interfaces and interactions, it was developed without graphic interfaces and has not been evaluated with heuristic evaluation or user testing yet. Its usability aspects cannot be concluded yet. However, from the perspective of development, and observations during the case study, some major limitations can already be spotted.

The solution is a prototype based on eye-tracking data collected with Tobii Pro Glasses 2 and pre-processed (i.e. fixation filtered) with Tobii Pro Lab. The structure of the input data is strictly constrained to the Tobii format. To be able to process data collected with other hardware models, processing modules (gaze filtering etc.) for different input formats are needed.

The prototype is currently implemented as Python command-line tools. Although the scripts are companied with documentation, it can still pose a challenge to users with little coding experience. The prototype also runs on many dependencies, which adds to the difficulties. For example, the Detectron2 framework used in the fixation mapping module (Section 4.3.2.2) originally built to run on macOS/Linux, and it cannot be directly installed on Windows machines without special configurations (changing the source code, using C++ build tools etc.). But on the other hand, Tobii Pro Lab requires Windows systems. This can cause a major inconvenience in using this prototype solution, especially among researchers without software development experience. The configuration of Amazon Web Services for the processing of think-aloud protocols can also be confusing. To make the prototype a usable analytical tool,

the dependency configurations need to be further simplified, the scripts need to be further encapsulated, and the modules need to be integrated into a graphic user interface.

6.4. Summary

In this chapter, the prototype was preliminarily evaluated using the data from the case study. The automated approach of mapping fixations to the object categories was consistent with the manual approach, and the cell phone was very accurately identified. The automated approach of mapping fixations to screen-contents was reasonably consistent with the manual approach: the automated approach was able to accurately identify some screens (i.e., the screens showing info and old maps), but inconsistency happened between similar-looking screens. The estimation of fixation screen coordinates was not very precise but generally agreed with the manually mapped results. The estimated fixation screen coordinates were not precise or accurate enough to pin-point the screen feature being viewed, but they could give a rough indication of which part of the screen is being viewed. The encoding of the chatbot was not very consistent with manual encoding and suffered from out-of-scope inputs, but it was able to identify a large proportion of the protocols describing usability issues.

The prototype has addressed the requirements and is able to assist the analysis of mobile eye-tracking data for GI user research by associating the visual attention with the environment and the representation on the mobile display, and assisting the linking the visual attention with the mental process with processed think-aloud protocols and location data. The case study has demonstrated the use of the prototype in an exploratory task, but it also has the potential to support more quantitative analysis. Potential future work includes mapping fixations with richer semantics, and further integration of location data and other data, such as screen-logging and motion sensors. Furthermore, to make the solution usable for users with different coding experience, the prototype can be further encapsulated and developed into integrated interfaces.

7. CONCLUSIONS

7.1. Summary of the thesis

Mobile eye-tracking has enabled GI user studies to be conducted in real-world environments to study the usability of mobile applications presenting spatio-temporal information and the cognitive process during the interaction with the information. But the dynamics in the real-world environments have posed challenges the effective analysis of the data, and the standard solutions provided by eye-tracker vendors don't necessarily fit the need of GI user research. This thesis addressed the problem by developing a prototype solution to help analyze mobile eye-tracking data collected (within a mixed-methods approach) for GI user research.

The implemented first-stage prototype solution (Chapter 4) consist of a fixation mapping component, a screen-recording processing component, and a think-aloud data processing component, and provided possibilities to synchronize the data and analyze them together. The fixation mapping component used panoptic segmentation to map fixations to real-world objects in the environments, it also estimated the screen-coordinates for fixations landing on the mobile display. The screen-recording processing component used histogram matching to help associate fixations with screen contents on the mobile display. The think-aloud data processing component transcribed, segmented, and encoded think-aloud protocols in a cloud workflow enabled by Amazon Web Services. The result from these components, with location data (GPS recordings), could be synchronized and analyzed together.

The prototype solution was demonstrated the GeoFARA case study as a proof-of-concept (Chapter 5). In the case study, mobile eye-tracking data, together with screen recording videos, think-aloud audios, and location data (GPS recordings) were processed and analyzed in an exploratory study that aimed to describe the process of fieldwork learning with GeoFARA and to find the usability issues of the application. The analysis explored the distribution and sequence of visual attention, identified usability issues from think-aloud protocols and described the process of fieldwork learning with GeoFARA with synchronized fixation-verbalization-location data. The preliminary evaluation with the case study data (Chapter 6) showed that automated approaches in the prototype solution was able to map fixations to real-world object categories and screen contents, and it could be used to identify usability issues from the think-aloud protocols.

The prototype has addressed the requirements and is able to assist the analysis of mobile eye-tracking data for GI user research by associating the visual attention with the environment and the representation on the mobile display, and assisting linking the visual attention with the mental process with processed think-aloud protocols and location data. Although it is only demonstrated with an exploratory task in the case study, it also has the potential to support more quantitative analysis.

7.2. Answering the research questions

1. What are the requirements for the solution in order to enable it to facilitate analyzing mobile eye-tracking data for GI science research following a mixed methods approach?

    - *What are the typical research questions being addressed with the help of mobile eye-tracking data in a mixed-methods approach and what kind of information is needed to answer those research questions?*

The typical research questions being addressed with the help of mobile eye-tracking in a mixed-methods approach are mainly related to the usability and design aspects of mobile application, and the cognitive process of spatial knowledge acquisition. They aim to describe the use and discover usability issues of a mobile application presenting spatial-temporal information; and to study the cognitive process as the user makes interactions between the environment, the representation of the environment and the mental map while performing a spatial task in the environment.

To answer those research questions, information on visual attention based on real-world objects and screen contents of the mobile display, mental process, and geographical context are needed. The visual attention includes the target, the duration, and the sequence of the attention. The target should include objects in the environment, as well as features or contents on the mobile display. The mental process refers to the test person's thoughts during the visual and physical behaviors. The geographical context refers to the context of the visual and physical behaviors and mental processes as test persons interact with the environment.

- *What is the current state-of-the-art analysis practice and what kind of information can be derived with it? What are the limitations of the current analytical solutions?*

Apart from manually inspecting the recording video to make discoveries, the current analysis practice of mobile eye-tracking data is centered around manually mapping fixations to reference images and defining AOIs on them. After the manual mapping, metrics (count, duration, revisit, etc.) are calculated with the mapped fixations and are further analyzed with qualitative or quantitative means. The processing and analysis of other data collected within the mixed-methods approach, in particular think-aloud data, are often carried out manually and independent of the analysis of eye-tracking data until their results are referred to each other.

Some major limitations of the current analytical solutions are: little support for automatic fixation mapping to real-world objects and object categories; no automated incorporation of screen contents on the mobile display into the analysis; not enough support for automated processing of think-aloud audio data and integrated analysis with mobile eye-tracking data; little support for the integration of other sensor data such as location data.

- *What additional functionalities are needed for an improved prototype solution in order to facilitate the analysis?*

An automated approach to associate fixations with semantic information on objects in the environment and screen features on the mobile display; automated approach to process think-aloud data and integrate it with the analysis of mobile eye-tracking data; integration of other data from the mixed-methods approach such as location data.

2. How can a prototype solution be designed in order to address the identified requirements?

Scene segmentation algorithms can be applied to directly attach object semantics to fixations. The panoptic segmentation model can be suitable because of its ability to coherently segment the image. For the mobile display, screen-recording videos can be processed to identify the screen content being viewed when a fixation is mapped to the screen.

The transcription of think-aloud audio can be done with available web services. Customizable chatbots services, supported by underlying natural language processing and understanding models, can act as a quick customizable off-the-shelf classifier for the encoding of the protocols.

The mapped fixations and processed protocols can then be synchronized with location data with their timestamps, and they can be explored and analyzed together with statistics or visualizations.

3. How can the prototype solution assist the analysis of mobile eye-tracking data to answer the relevant research questions?

   - *What information can be extracted with the prototype solution and what is its advantage in extracting* the *information comparing to existing analytical solutions?*

   The distribution and the sequence of visual attention (e.g. fixation-based metrics) can be extracted more easily without manual fixation mapping. It's easier to know what object being looked at and how the environment and the mobile display are visually explored and linked. With the processed think-aloud protocols, usability issues can be discovered more easily without manually going through the entire transcript. And when the protocols are synchronized with mapped fixations and explored simultaneously, they can directly support the patterns and discoveries in the fixation data and provide insights on the test person's mental process. The synchronized location data adds the spatial and environmental context into the analysis of fixations and verbalizations, and they can be explored and analyzed together.

   The main advantages of the prototype solution are automation of fixation mapping and think-aloud data processing and possibilities for synchronized analysis of data collected by the mixed-methods approach.

   - *How can the prototype solution be used in the analysis of mobile eye-tracking data to answer the relevant research questions?*

   The prototype solution can help to discover and describe the interaction between the reality, the representation and the mental map by automizing some of the laborious operations and integrating the data from the mixed-methods approach. For exploratory and qualitative analysis, it can produce summary statistics and visualizations that provide clues on potential patterns and unique spots in the mobile eye-tracking data, and reduce the amount of manual inspection of the recording videos; it can support the exploration of synchronized fixation-verbalization-location data to provide a more comprehensive view of visual and physical behaviors, and mental processes as people interact with applications presenting the spatio-temporal information or solve a spatial task in the real-world environment. For more quantitative analysis, metrics can be calculated from the data without laborious manual work.

7.3. Further testing, development, and research

The current first-stage prototype can be further tested and developed. It needs to be further tested with different experiment setups, including different environments (especially the indoor scenario), different kinds of screen contents, and different tasks for the test persons. It should also be tested and evaluated with different analytical tasks (e.g. quantitative between-group analysis, encoding protocols with different coding schemes) performed by real researchers. Improvements in compatibility with different eye-trackers and data formats is also needed: modules that process raw gaze data is needed to make it compatible with mobile eye-tracking data collected with other hardware in different data formats. To make the prototype a usable analytical tool, the dependency configurations need to be further simplified, the scripts need to be further encapsulated, and graphic interfaces can be developed.

Further research in such analytical solutions can consider fixation (gaze) mapping, think-aloud processing and the integration of data in the mixed-methods approach. Fixations need to be mapped with richer semantics to make the automated mapping more comparable to human scene understanding during manual mapping. The integration of object-based and featuring-matching fixation mapping methods can be explored. Methods to improve the estimation of fixation screen coordinates and to identify screen features being viewed can also be explored. (Semi-)automated encoding of the protocols can be further explored, both with state-of-art natural language processing and understanding models and emerging off-the-shelf services. More complex encoding tasks can also be explored, for example, encoding with context information. More methods can be included in the mixed-methods approach and be integrated into the analysis of mobile eye-tracking data, including but not limited to action logging and motion sensing. The resulting data might also be used for data-driven analysis.

LIST OF REFERENCES

Amati, M., Ghanbari Parmehr, E., McCarthy, C., & Sita, J. (2018). How eye-catching are natural features when walking through a park? Eye-tracking responses to videos of walks. *Urban Forestry and Urban Greening*, *31*, 67–78. https://doi.org/10.1016/j.ufug.2017.12.013

Amazon Web Services. (2019). Find Utterance Conflicts in Your Model | Alexa Skills Kit. Retrieved May 19, 2020, from https://developer.amazon.com/en-GB/docs/alexa/custom-skills/find-utterance-conflicts-in-your-model.html

Amazon Web Services. (2020a). AMAZON.FallbackIntent - Amazon Lex. Retrieved May 24, 2020, from https://docs.aws.amazon.com/lex/latest/dg/built-in-intent-fallback.html

Amazon Web Services. (2020b). *Amazon Lex Developer Guide*.

Amazon Web Services. (2020c). *Amazon Transcribe Developer Guide*.

Amazon Web Services. (2020d). Managing Conversation Context - Amazon Lex. Retrieved May 26, 2020, from https://docs.aws.amazon.com/lex/latest/dg/context-mgmt.html

Amazon Web Services. (2020e). Quotas - Amazon Lex. Retrieved May 24, 2020, from https://docs.aws.amazon.com/lex/latest/dg/gl-limits.html

Aujogue, J. B., & Aussem, A. (2019). Hierarchical Recurrent Attention Networks for Context-Aware Education Chatbots. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 2019-July). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/IJCNN.2019.8852445

Bauer, C., & Ludwig, B. (2019). Schematic maps and indoor wayfinding. *Leibniz International Proceedings in Informatics, LIPIcs*, *142*(23), 1–14. https://doi.org/10.4230/LIPIcs.COSIT.2019.23

Benjamins, J. S., Hessels, R. S., & Hooge, I. T. C. (2018). GazeCode: Open-source software for manual mapping of mobile eye-tracking data. *Eye Tracking Research and Applications Symposium (ETRA)*. https://doi.org/10.1145/3204493.3204568

Blascheck, T., Kurzhals, K., Raschke, M., Burch, M., Weiskopf, D., & Ertl, T. (2017). Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum*, *36*(8), 260–284. https://doi.org/10.1111/cgf.13079

Brügger, A., Richter, K.-F., & Fabrikant, S. I. (2017). Which egocentric direction suffers from visual attention during aided wayfinding? *Eye Tracking for Spatial Research, Proceedings of the 3rd International Workshop*. https://doi.org/10.3929/ETHZ-B-000222472

Brügger, A., Richter, K. F., & Fabrikant, S. I. (2019). How does navigation system behavior influence human behavior? *Cognitive Research: Principles and Implications*, *4*(1). https://doi.org/10.1186/s41235-019-0156-5

Caesar, H., Uijlings, J., & Ferrari, V. (2016). COCO-Stuff: Thing and Stuff Classes in Context. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1209–1218. Retrieved from http://arxiv.org/abs/1612.03716

Çöltekin, A., Fabrikant, S. I., & Lacayo, M. (2010). Exploring the efficiency of users' visual analytics strategies based on sequence analysis of eye movement recordings. *International Journal of Geographical Information Science*, *24*(10), 1559–1575. https://doi.org/10.1080/13658816.2010.511718

De Cock, L., Viaene, P., Ooms, K., Michels, R., De Wulf, A., Vanhaeren, N., … De Maeyer, P. (2019). Comparing written and photo-based indoor wayfinding instructions through eye fixation measures and user ratings as mental effort assessments. *Journal of Eye Movement Research*, *12*(1), 1–14. https://doi.org/10.16910/jemr.12.1.1

De Tommaso, D., & Wykowska, A. (2019). TobiiglassespySuite: An open-source suite for using the Tobii Pro Glasses 2 in eye-tracking studies. *Eye Tracking Research and Applications Symposium (ETRA)*. https://doi.org/10.1145/3314111.3319828

Delikostidis, I. (2011). *Improving the usability of pedestrian navigation systems (doctoral dissertation)*. University of Twente.

Dong, W., Wang, S., Chen, Y., & Meng, L. (2018). Using Eye Tracking to Evaluate the Usability of Flow Maps. *ISPRS International Journal of Geo-Information*, *7*(7), 281. https://doi.org/10.3390/ijgi7070281

Ericsson, A., & Simon, H. A. (Herbert A. (1993). *Protocol analysis : verbal reports as data*. MIT Press.

Fischer, B., & Ramsperger, E. (1984). *Experimental Brain Research Human express saccades: extremely short reaction times of goal directed eye movements*. *Exp Brain Res* (Vol. 57).

Franke, C., & Schweikart, J. (2016). Investigation of landmark-based pedestrian navigation processes with a mobile eye tracking system. In G. Gartner & H. Huang (Eds.), *Progress in Location-Based Services*

2016. *Lecture Notes in Geoinformation and Cartography* (pp. 105–130). Kluwer Academic Publishers. https://doi.org/10.1007/978-3-319-47289-8_6
Franke, C., & Schweikart, J. (2017). Investigation of Landmark-Based Pedestrian Navigation Processes with a Mobile Eye Tracking System. In Georg Gartner & H. Huang (Eds.), *Progress in Location-Based Services 2016* (pp. 105–130). Cham: Springer International Publishing.
Gerring, J. (2004). What is a case study and what is it good for? *American Political Science Review*, *98*(2), 341–354. https://doi.org/10.1017/S0003055404001182
Göbel, F., Kiefer, P., & Raubal, M. (2019). FeaturEyeTrack: automatic matching of eye tracking data with map features on interactive maps. *GeoInformatica*, *23*(4), 663–687. https://doi.org/10.1007/s10707-019-00344-3
Gog, T. van, Kester, L., Nievelstein, F., Giesbers, B., & Paas, F. (2009). Uncovering cognitive processes: Different techniques that can contribute to cognitive load research and instruction. *Computers in Human Behavior*, *25*(2), 325–331. https://doi.org/10.1016/j.chb.2008.12.021
Goldberg, J. H., & Kotval, X. P. (1999). Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, *24*(6), 631–645. https://doi.org/10.1016/S0169-8141(98)00068-7
Google Cloud. (2020). Assessing the quality of training phrases in Dialogflow intents. Retrieved May 26, 2020, from https://cloud.google.com/solutions/assessing-the-quality-of-training-phrases-in-dialogflow-intents
Haklay, M. M. (2010). *Interacting with Geospatial Technologies*. *Interacting with Geospatial Technologies*. https://doi.org/10.1002/9780470689813
He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969). https://doi.org/10.1109/TPAMI.2018.2844175
Herlitz, M. (2018). Analyzing the Tobii Real-world- mapping tool and improving its workflow using Random Forests, 42.
Hollander, J. B., Purdy, A., Wiley, A., Foster, V., Jacob, R. J. K., Taylor, H. A., & Brunyé, T. T. (2019). Seeing the city: using eye-tracking technology to explore cognitive responses to the built environment. *Journal of Urbanism: International Research on Placemaking and Urban Sustainability*, *12*(2), 156–171. https://doi.org/10.1080/17549175.2018.1531908
Jones, C. E., & Weber, P. (2012). Towards Usability Engineering for Online Editors of Volunteered Geographic Information: A Perspective on Learnability. *Transactions in GIS*, *16*(4), 523–544. https://doi.org/10.1111/j.1467-9671.2012.01319.x
Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*, *8*(4), 441–480. https://doi.org/10.1016/0010-0285(76)90015-3
Kiefer, P., Giannopoulos, I., & Raubal, M. (2014). Where am i? Investigating map matching during self-localization with mobile eye tracking in an urban environment. *Transactions in GIS*, *18*(5), 660–686. https://doi.org/10.1111/tgis.12067
Kiefer, P., Giannopoulos, I., Raubal, M., & Duchowski, A. (2017). Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation*, *17*(1–2), 1–19. https://doi.org/10.1080/13875868.2016.1254634
Kiefer, P., Straub, F., & Raubal, M. (2011). Towards Location - Aware Mobile Eye Tracking, 313–316.
Kiefer, P., Straub, F., & Raubal, M. (2012). Location-Aware Mobile Eye Tracking for the Explanation of Wayfinding Behavior. *Proceedings of the AGILE'2012 International Conference on Geographic Information Science*, (May 2014).
Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2018). Panoptic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2019-June*, 9396–9405. Retrieved from http://arxiv.org/abs/1801.00868
Koletsis, E., van Elzakker, C. P. J. M., Kraak, M.-J., Cartwright, W., Arrowsmith, C., & Field, K. (2017). An investigation into challenges experienced when route planning, navigating and wayfinding. *International Journal of Cartography*, *3*(1), 4–18. https://doi.org/10.1080/23729333.2017.1300996
Konopka, M. (2019). GazeToolkit: Toolkit for processing eye movement data, fixation filtering, smoothing, etc. Retrieved from https://github.com/uxifiit/GazeToolkit
Kowler, E. (2011, July 1). Eye movements: The past 25years. *Vision Research*. Pergamon. https://doi.org/10.1016/j.visres.2010.12.014
Krassanakis, V., & Cybulski, P. (2019). A review on eye movement analysis in map reading process: the status of the last decade. *GEODESY AND CARTOGRAPHY© Polish Academy of Sciences*, *68*(1), 191–209. https://doi.org/10.24425/gac.2019.126088

Kurzhals, K., Hlawatsch, M., Seeger, C., & Weiskopf, D. (2017). Visual Analytics for Mobile Eye Tracking. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 301–310. https://doi.org/10.1109/TVCG.2016.2598695

Lander, C., Herbig, N., Löchtefeld, M., Wiehr, F., & Krüger, A. (2017). Inferring landmarks for pedestrian navigation from mobile eye-tracking data and Google Street View. *Conference on Human Factors in Computing Systems - Proceedings*, *Part F1276*, 2721–2729. https://doi.org/10.1145/3027063.3053201

Larson, S., Mahendran, A., Peper, J. J., Clarke, C., Lee, A., Hill, P., … Mars, J. (2019). An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 1311–1316). Association for Computational Linguistics. Retrieved from https://dialogflow.com

Li, Q. (2017). *Use of Maps in Indoor Wayfinding*. University of Twente. Retrieved from https://library.itc.utwente.nl/login/2017/msc/gfm/qli.pdf

Liao, H., Dong, W., Huang, H., Gartner, G., & Liu, H. (2019). Inferring user tasks in pedestrian navigation from eye movement data in real-world environments. *International Journal of Geographical Information Science*, *33*(4), 739–763. https://doi.org/10.1080/13658816.2018.1482554

Liao, H., Dong, W., Peng, C., & Liu, H. (2017). Exploring differences of visual attention in pedestrian navigation when using 2D maps and 3D geo-browsers. *Cartography and Geographic Information Science*, *44*(6), 474–490. https://doi.org/10.1080/15230406.2016.1174886

MacDonald, C. M., & Atwood, M. E. (2013). Changing Perspectives on Evaluation in HCI: Past, Present, and Future. *Conference on Human Factors in Computing Systems - Proceedings*, *2013-April*, 1969–1978. https://doi.org/10.1145/2468356.2468714

Macinnes, J. J., Iqbal, S., Pearson, J., & Johnson, E. N. (2018). Mobile Gaze Mapping: A Python package for mapping mobile gaze data to a fixed target stimulus Software • Review • Repository • Archive. *Journal of Open Source Software*, *3*(31), 984. https://doi.org/10.21105/joss.00984

Merino, G. S. A. D., Riascos, C. E. M., Costa, A. D. L., Elali, G. V. M. de A., & Merino, E. (2018). The focus of visual attention in people with motor disabilities through Eye tracking. *Gestão & Tecnologia de Projetos*, *13*(3), 7–20. https://doi.org/10.11606/gtp.v13i3.146091

Metallinou, A. (2018). Amazon Scientists Use Transfer Learning to Accelerate Development of New Alexa Capabilities. Retrieved March 1, 2020, from https://www.amazon.science/blog/amazon-scientists-use-transfer-learning-to-accelerate-development-of-new-alexa-capabilities

Müller, P., Buschek, D., Huang, M. X., & Bulling, A. (2019). Reducing calibration drift in mobile eye trackers by exploiting mobile phone usage. In *Eye Tracking Research and Applications Symposium (ETRA)*. Association for Computing Machinery. https://doi.org/10.1145/3314111.3319918

Niehorster, D. C., Hessels, R. S., & Benjamins, J. S. (2020). GlassesViewer: Open-source software for viewing and analyzing data from the Tobii Pro Glasses 2 eye tracker. *Behavior Research Methods*. https://doi.org/10.3758/s13428-019-01314-1

Ogawa, M., & Aizawa, K. (2019). Identification of Buildings in Street Images Using Map Information. In *Proceedings - International Conference on Image Processing, ICIP* (Vol. 2019-September, pp. 984–988). IEEE Computer Society. https://doi.org/10.1109/ICIP.2019.8803066

Ohm, C., Müller, M., & Ludwig, B. (2017). Evaluating indoor pedestrian navigation interfaces using mobile eye tracking. *Spatial Cognition and Computation*, *17*(1–2), 89–120. https://doi.org/10.1080/13875868.2016.1219913

Olsen, A. (2012). *The Tobii I-VT Fixation Filter: Algorithm description*. *Tobii Technology*. Retrieved from https://www.tobiipro.com/siteassets/tobii-pro/learn-and-support/analyze/how-do-we-classify-eye-movements/tobii-pro-i-vt-fixation-filter.pdf

Ooms, K., Coltekin, A., De Maeyer, P., Dupont, L., Fabrikant, S., Incoul, A., … Van der Haegen, L. (2015). Combining user logging with eye tracking for interactive and dynamic applications. *Behavior Research Methods*, *47*(4), 977–993. https://doi.org/10.3758/s13428-014-0542-3

Plotly Technologies Inc. (2015). Collaborative data science. Montréal, QC: Plotly Technologies Inc. Retrieved from https://plot.ly

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A., Hall, A., … van Elzakker, C. P. J. M. (2017). User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, *3*(sup1), 61–89. https://doi.org/10.1080/23729333.2017.1288534

Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the Eye Tracking Research and Applications Symposium 2000*, 71–78.

https://doi.org/10.1145/355017.355028
Schnitzler, V., Giannopoulos, I., Hölscher, C., & Barisic, I. (2016). The interplay of pedestrian navigation, wayfinding devices, and environmental features in indoor settings. *Eye Tracking Research and Applications Symposium (ETRA)*, *14*, 85–93. https://doi.org/10.1145/2857491.2857533
SensoMotoric Instruments GmbH. (2017). BeGaze 3.7 Manual.
TensorFlow. (2015). TensorFlow code style guide. Retrieved May 17, 2020, from https://www.tensorflow.org/community/contribute/code_style
Tobii Pro. (2015a). Eye tracking software and devices - Products by Tobii Pro.
Tobii Pro. (2015b). Real-World Mapping Tool for Pro Glasses 2 - YouTube. Retrieved March 3, 2020, from https://www.youtube.com/watch?v=JU4m6UsfnQY
Tobii Pro. (2015c). Types of eye movements. Retrieved March 3, 2020, from https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/types-of-eye-movements/
Tobii Pro. (2016). *Tobii Pro Glasses 2 User's Manual. Tobii Pro Glasses 2* (Vol. 46). Retrieved from http://www.tobiipro.com/product-listing/tobii-pro-glasses-2/
Tobii Pro. (2019a). Automatic mapping in Tobii Pro Lab is not working as expected. What can I do? Retrieved March 3, 2020, from https://connect.tobiipro.com/s/article/Automatic-mapping-in-Tobii-Pro-Lab-is-not-working-as-expected-What-can-I-do?language=en_US
Tobii Pro. (2019b). *Tobii Pro Lab User's Manual*. Retrieved from www.tobiipro.com
Tomasi, M., Pundlik, S., Bowers, A. R., Peli, E., & Luo, G. (2016). Mobile gaze tracking system for outdoor walking behavioral studies. *Journal of Vision*, *16*(3). https://doi.org/10.1167/16.3.27
Utebaliyeva, M. (2019). *The Use of Maps on Smartwatches*. University of Twente.
van Elzakker, C. P. J. M., & Wealands, K. (2007). Use and users of multimedia cartography. *Multimedia Cartography: Second Edition*, 487–504. https://doi.org/10.1007/978-3-540-36651-5_34
Viaene, P., Ooms, K., Vansteenkiste, P., Lenoir, M., & De Maeyer, P. (2014). The use of eye tracking in search of indoor landmarks. *CEUR Workshop Proceedings*, *1241*, 52–56.
Viaene, P., Vanclooster, A., Ooms, K., & De Maeyer, P. (2015). Thinking aloud in search of landmark characteristics in an indoor environment. *2014 Ubiquitous Positioning Indoor Navigation and Location Based Service, UPINLBS 2014 - Conference Proceedings*, 103–110. https://doi.org/10.1109/UPINLBS.2014.7033716
Viaene, P., Vansteenkiste, P., Lenoir, M., De Wulf, A., & De Maeyer, P. (2016). Examining the validity of the total dwell time of eye fixations to identify landmarks in a building. *Journal of Eye Movement Research*, *9*(3), 1–11. https://doi.org/10.16910/jemr.9.3.4
Wan, Q., Kaszowska, A., Panetta, K., A Taylor, H., & Agaian, S. (2019). A Comprehensive Head-mounted Eye Tracking Review: Software Solutions, Applications, and Challenges. *Electronic Imaging*, *2019*(3), 654-1-654–659. https://doi.org/10.2352/issn.2470-1173.2019.3.sda-654
Wang, C., Chen, Y., Zheng, S., & Liao, H. (2019). Gender and age differences in using indoor maps for wayfinding in real environments. *ISPRS International Journal of Geo-Information*, *8*(1). https://doi.org/10.3390/ijgi8010011
Wang, X. (2018). GeoFARA on Vimeo. Retrieved April 13, 2020, from https://vimeo.com/263052616
Wang, X. (2020). *User-Centered Design of A Mobile Application Using A Combination of Augmented Reality and Maps for Geo-Fieldwork Education*. University of Twente.Manualscript in preparation.
Wang, X., van Elzakker, C. P. J. M., & Kraak, M.-J. (2017). Conceptual design of a mobile application for geography fieldwork learning. *ISPRS International Journal of Geo-Information*, *6*(11). https://doi.org/10.3390/ijgi6110355
Wang, X., van Elzakker, C. P. J. M., Kraak, M.-J., & Köbben, B. (2017). GeoFARA: design, development and evaluation of a mobile human geography fieldwork application: powerpoint. In *28th International Cartographic Conference, ICC 2017 - Washington, United States*.
Weber, M., Luiten, J., & Leibe, B. (2019). Single-Shot Panoptic Segmentation. *ArXiv Preprint*. Retrieved from http://arxiv.org/abs/1911.00764
Wenczel, F., Hepperle, L., & von Stülpnagel, R. (2017). Gaze behavior during incidental and intentional navigation in an outdoor environment. *Spatial Cognition and Computation*, *17*(1–2), 121–142. https://doi.org/10.1080/13875868.2016.1226838
Wolf, J., Hess, S., Bachmann, D., Lohmeyer, Q., & Meboldt, M. (2018). Automating areas of interest analysis in mobile eye tracking experiments based on machine learning. *Journal of Eye Movement Research*, *11*(6), 6. https://doi.org/10.3929/ETHZ-B-000309840
Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified Perceptual Parsing for Scene Understanding. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11209 LNCS*, 432–448. Retrieved from http://arxiv.org/abs/1807.10221

Yang, S. C. (2003). Reconceptualizing think-aloud methodology: Refining the encoding and categorizing techniques via contextualized perspectives. *Computers in Human Behavior*, *19*(1), 95–115. https://doi.org/10.1016/S0747-5632(02)00011-0

Zagermann, J., Pfeil, U., & Reiterer, H. (2016). Measuring cognitive load using eye tracking technology in visual computing. *ACM International Conference Proceeding Series*, *24-October*, 78–85. https://doi.org/10.1145/2993901.2993908

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2016). Semantic Understanding of Scenes through the ADE20K Dataset. *International Journal of Computer Vision*, *127*(3), 302–321. Retrieved from http://arxiv.org/abs/1608.05442

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., & Torralba, A. (2019). Semantic Understanding of Scenes Through the ADE20K Dataset. *International Journal of Computer Vision*, *127*(3), 302–321. https://doi.org/10.1007/s11263-018-1140-0

APPENDIX A CODE REPOSITORIES AND INSTRUCTIONS (README) FOR THE PROTOTYPE SOLUTION

The source code and (mini) examples can be found in the following repositories:

- Automated fixation mapping with screen recording processing: https://github.com/myhjiang/et_mapping
- Think-aloud processing with Amazon Web Services: https://github.com/myhjiang/aws_ta
- Synchronization and mini examples for integrated analysis: https://github.com/myhjiang/mobile_et_example

A copy of the instructions (readme files can be found in the following pages. It's recommended to access them via the repositories (to better access inline links and potential changes in the future).

The following pages contain:

- Readme: Fixation Mapping for Mobile Eye-Tracking: to Real-World Objects and Screen Contents
- Readme: Simple think-aloud data processing with Amazon Web Services
- Readme: Mini Examples for Analyzing Mobile Eye-tracking Data In A Mixed-Methods Approach

Fixation Mapping for Mobile Eye-Tracking: to Real-World Objects and Screen Contents

A little tool to process mobile eye-tracking data. It automatically maps fixations to real-world objects and screen contents (when a mobile device is involved in the experiment).

It maps fixations to real-world objects with panoptic segmentation (using FAIR's Detectron2 framework). If a fixation is mapped to object "cell phone", it can be further linked to screen contents on the phone with the help of screen recording videos (content-based image searching).

Note: the current version only supports fixation data collected with Tobii Pro Glasses 2 and exported by Tobii Pro Lab.

This is a part of MSc. Thesis *A Solution to Analyze Mobile Eye-tracking Data for User Research in GI Science* by Yuhao (Markie) Jiang, ITC-University of Twente, Enschede, The Netherlands, June 2020.

Requirements

- Detectron2

  For Linux / MacOS: follow the official guide for installation and dependencies.

  For windows (unfortunately), follow the build procedure in this repo to build Detectron2 on your machine.
- cv2, version 3.4.2
- numpy
- pandas

The scripts are developed and tested on Python 3.7

Prepare the data

Eye-tracking Video

The **full stream (i.e. not segment)** video of the participant. This video can be acquired from the live data recorder as fullstream.mp4, or by exporting the entire video from Tobii ProLab.

Fixation data table

`.tsv` data table exported with Tobii ProLab. Must include the following columns: `Recording timestamp, Eye movement type, Eye movement type index, Gaze event duration, Fixation point X, Fixation point Y`. Other columns are optional.

The fixation data table can be the the data of a segment.

Raw gaze data is not supported in this version.

Screen recording video and candidate screenshots

The video of screen recording (if applicable) recorded during the experiment.

The candidate screenshots are the "example" screen contents that you want to compare the frames of screen recording with. It should be the ones that are typical, or the ones of special interest. They should be stored in a folder. `.jpg` and `.png` formats are accepted.

Mapping fixations to real-world objects

Run fixation_mapping.py

Before running, make sure that the `labels` folder is in the same working directory of the script.

Arguments:

- Detectron (dp): path to your Detectron2 folder. This is usually in your virtual environment if you use one, or could be in your `C:\Users` if you have installed Detectron2 globally.
- VideoFile (vf): path to the full-stream video file.
- DataFile (ff): path to fixation data table (.tsv)
- OutFolder (of): folder for output file

For example: `$ python fixation_mapping.py C:\\Users\\admin\\detectron2 F:\\videos\\recording30_full.mp4 F:\\exports\\recording30_segment.tsv F:\\exports\\mapped`

Note: During the (first time) execution of the script, a `.pkl` file will be downloaded as the weight for the segmentation model. It will be saved at `./model/model_final_dbfeb4.pkl`

Output

A `.tsv` data table with each fixation mapped to an object. An example table would look like:

Recording timestamp	Eye movement type index	Gaze event duration	Fixation point X	Fixation point Y	target	phone_x	phone_y
715321	1514	60	765	357	road		
715521.5	1515	300	559	308	road		
715762.5	1516	140	632	355	car		
716361.5	1517	580	1021	700	cell phone	0.76	0.33

where `target` columns record the object the fixation is mapped to. A complete list of the available objects can be found at `labels/complete_list.txt`

When a fixations is mapped to a `cell phone` object, the approximate (proportional) location of the fixation on the cell phone is estimated and stored in `phone_x` and `phone_y` columns.

Syncing mobile eye-tracking data with screen-recordings of the mobile stimuli

Run screen_recording_match.py

Before running, make sure that the `imsearch` folder is in the same working directory of the script.

Arguments

- TargetImagesFolder (im): path to the folders with candidate images
- ScreenVideo (sr): path to the screen recording video
- FixationsFile (fx): path to **mapped** fixation file (`.tsv`)

- TimeOffset (t): time offset in milliseconds (i.e., the time difference between the start of screen-recording video and the start of eye-tracking recording)
- optional: --index (-i): path to an existing index file. If index is already built, use the old index for faster processing time
- optional: --binsize (-b): customize HSV bin size for histogram comparison. Default bin size: (20, 20, 30), 20 bins for H, 20 bins for S and 30 bins for V.

For example `$ python screen_recording_match.py "F:/videos/image_pool" "F:/videos/screen.mp4" "F:/exports/mapped/tp1_fixaton_mapped.tsv" 201839`

Output

A `.tsv` data table, when the `target` of the fixation is `cell phone`, one `best_match` image (candidate) will be assigned to the fixation. The output file is saved to the same folder of the input fixation data table. An example table would look like:

Recording timestamp	Eye movement type index	Gaze event duration	Fixation point X	Fixation point Y	target	phone_x	phone_y	best_match
715321	1514	60	765	357	road			
715521.5	1515	300	559	308	road			
715762.5	1516	140	632	355	car			
716361.5	1517	580	1021	700	cell phone	0.76	0.33	info_park.jpg

What's next

See my other repo for some mini-examples of analyzing the data.

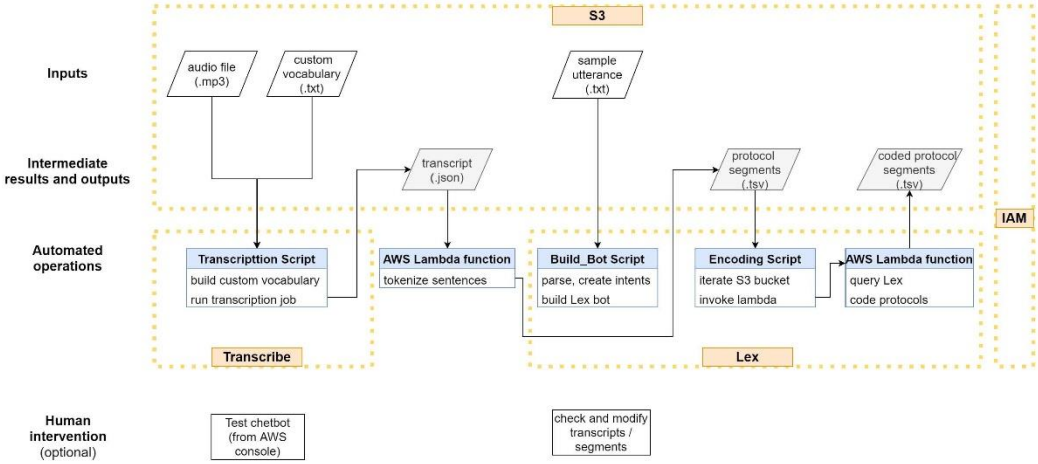
Sources and more

- Detectron2
- Panoptic segmentation (paper)
- COCO panoptic dataset (API) and COCO panoptic task description 2019
- Tobii ProLab manual

Simple think-aloud data processing with Amazon Web Services

An almost automated workflow to process think-aloud data powered by Amazon Web Services (S3, Lambda, Transcribe, Lex).

The simple workflow: transcription -> segmentation -> encoding



This is a part of MSc. Thesis *A Solution to Analyze Mobile Eye-tracking Data for User Research in GI Science* by Yuhao (Markie) Jiang, ITC-University of Twente, Enschede, The Netherlands, June 2020.

Requirements

- AWS account
- Configured AWS CLI
- Boto3

Languages

AWS Transcribe supports multiple languages, but Lex only supports English. Thus this whole workflow only works for English (US), while the transcription part can be used for other languages supported by Amazon.

Prepare the data

All inputs files should be stored in an S3 bucket with the following structure:

```
S3 Bucket
├── intents
|   └── code_name.txt
|   └── code_name.txt
|   └── ...
├── custom_vocabulary.txt
├── think_aloud_audio.mp3
├── another_audio.mp3
├── ...
```

`custom_vocabulary.txt` should be in the Table Format

Audio files can be: `.mp3, .mp4, .wav, .flac`

`intents` contains the `.txt` files

Note: Intents don't have to be in the same buckets as the audio and vocabulary files, but they should be all stored under the sub-folder of `intents`
Intents should follow the naming of `code_name.txt`, where `code` is a simple code you want to use for your intent, and `name` is the name of the intent. For example, `I_mapinteraction.txt` represents an intent called "mapinteraction" with the code I.
The intent file should contain sample utterances separated by line break.
For example, the `I_mapinteraction.txt` has the following content:

```
zoom in.
I'll zoom out a little bit more.
I'm click the button to...
...
```

Some AWS access configuration

You will need your AWS region name to run `transcribe.py`
You will also need a IAM role that has the following policies attached:

- AWSLambdaExecute
- AWSLambdaBasicExecutionRole
- AmazonLexRunBotsOnly

This can be done via AWS CLI or from the IAM console.

Note: `transcribe.py` will create a temporary data-access-role that allow Transcribe to read and write into your S3 bucket when jobs are queued. This role is attached with these policies provided by Amazon. This role will be deleted as the script finishes.

Run

Make sure the `lambda_function` folder is in the same directory as the scripts before running the scripts!

transcribe.py

Arguments:

- InputBucket (in): name of the bucket where you store your custom vocabulary and audio files
- OutputBucket (out): name of the bucket to store transcripts, can be the same with InputBucket
- Region (rg): the region of your CLI, **has to be the same with the region of the buckets!**
- Role (rl): the IAM role that has the accesses mentioned above.

For example: `$ python transcribe.py bucket1 bucket2 us-west-2, FullAccessRole`

build_bot_s3.py

Arguments:

- Bucket (b): name of the bucket where the intent folder is stored.
- BotName (n): name of the bot

For example: `$ python build_bot_s3.py mybucket mybot`

encode.py

Arguments:

- InputBucket (in): name of the bucket that stores the transcript segments produced by `transcribe.py`
- Role (rl): the same role name used for `transcribe.py`
- BotName (bn): name of the Lex bot
- BotAlias (ba): alias name of the Lex bot. If the bot is created with `build_bot_s3.py`, then the alias is `[botname]_alias`

For example: `$ python encode.py mybucket FullAccessRole mybot mybot_alias`

Outputs and intermediate results

`transcribe.py` will write transcripts in `.json` as the AWS default format, as well as segmented transcripts (by sentence) in `.tsv` to your output S3 bucket. Segmented transcripts follow the format of:

start_time	end_time	content
2.11	7.17	and therefore I also need to old map

`build_bot_s3.py` will create intents and build a Lex chatbot. You can test, modify and rebuild the bot at the Lex Console.

`encode.py` will write coded transcript sentences to your S3 bucket as `tsv` files. They follow the format of:

start_time	end_time	content	BotCode
2.11	7.17	and therefore I also need to old map	A

Human intervention

Amazon is quite smart but not perfect. Thus human intervention is recommended as intermediate results are produced.

Modifying the transcripts: you can modify the tsv transcript segments by downloading and editing it in your text/sheet editor, or you can use this tool to modify the json transcript. Once you upload the modified json to the same bucket, an new segment file will be automatically generated.

Testing the bot and modifying the intents: this is easiest done through the Lex console. Do not run the build bot script again after you have created intents or bots with the same name.

If you also do mobile eye-tracking

See my other repo for some mini-examples of analyzing protocols together with mobile eye-tracking data

Sources and AWS docs

- Amazon Transcribe
- Amazon Lex
- Amazon S3 Simple Storage
- Amazon IAM

Mini Examples for Analyzing Mobile Eye-tracking Data In A Mixed-Methods Approach

Some mini examples of an exploratory analysis for mobile eye-tracking data collected in a mixed-methods approach for GI user research.

This is a part of MSc. Thesis *A Solution to Analyze Mobile Eye-tracking Data for User Research in GI Science* by Yuhao (Markie) Jiang, ITC-University of Twente, Enschede, The Netherlands, June 2020

Analysis based on

- my automated fixation mapping tool
- my think-aloud processing tool
- Plot.ly

What it does

- processing: fixation mapping to real-world objects and screen contents
- visualizing the mapped fixations: distribution and sequence
- synchronizing mapped fixations with verbalizations and location data
- visualizing fixations together with verbalizations and location data.

Data

For file size limitations, the data folder does not contain video / audio data.

The data folder contains:

- `export_fixation.tsv` fixation file exported with Tobii Pro Lab
- `mapped_fixation.tsv`, `replaced_mapped_fixation.tsv`, `screen_replaced_mapped_fixation.tsv`, `grouped_screen_replaced_mapped_fixation.tsv` mapped fixations, outputs and processed outputs from my automated fixation mapping tool
- `gps.csv` GPS measurements along the segment
- `segment_protocol.tsv` transcribed think-aloud protocol segments
- `sync_fixation_protocol.tsv`, `sync_fixation_gps.tsv`, `sync_protocol_gps` synchronized files

Notebooks

- `visual_attention.ipynb` exploring the distribution and sequence of visual attention
- `fixation_protocol.ipynb` exploring fixations together with protocols
- `location_combined.ipynb` exploring fixations and protocols together with location data

Detailed descriptions can be found within the notebooks. ***GitHub cannot show Plot.ly interactive plots properly in Notebooks. It's recommended to use those notebooks locally***

Notebook dependencies

- plotly (go, express and io)
- pandas
- numpy
- re
- bisect

Sample plots

Sample plots can be found in folder `plots`.

An example of a fixation sequence with protocols on a map:

APPENDIX B CONFIGURATION DETAILS OF DETECTRON2 PANOPTIC SEGMENTATION MODEL

Model: panoptic_fpn_R_50_1x

Model metrics url: https://dl.fbaipublicfiles.com/detectron2/COCO-PanopticSegmentation/panoptic_fpn_R_50_1x/139514544/metrics.json

Model configuration:


```
CUDNN_BENCHMARK: False
DATALOADER:
  ASPECT_RATIO_GROUPING: True
  FILTER_EMPTY_ANNOTATIONS: True
  NUM_WORKERS: 4
  REPEAT_THRESHOLD: 0.0
  SAMPLER_TRAIN: TrainingSampler
DATASETS:
  PRECOMPUTED_PROPOSAL_TOPK_TEST: 1000
  PRECOMPUTED_PROPOSAL_TOPK_TRAIN: 2000
  PROPOSAL_FILES_TEST: ()
  PROPOSAL_FILES_TRAIN: ()
  TEST: ('coco_2017_val_panoptic_separated',)
  TRAIN: ('coco_2017_train_panoptic_separated',)
GLOBAL:
  HACK: 1.0
INPUT:
  CROP:
    ENABLED: False
    SIZE: [0.9, 0.9]
    TYPE: relative_range
  FORMAT: BGR
  MASK_FORMAT: polygon
  MAX_SIZE_TEST: 1333
  MAX_SIZE_TRAIN: 1333
  MIN_SIZE_TEST: 800
  MIN_SIZE_TRAIN: (640, 672, 704, 736, 768, 800)
  MIN_SIZE_TRAIN_SAMPLING: choice
MODEL:
  ANCHOR_GENERATOR:
    ANGLES: [[-90, 0, 90]]
    ASPECT_RATIOS: [[0.5, 1.0, 2.0]]
    NAME: DefaultAnchorGenerator
    SIZES: [[32], [64], [128], [256], [512]]
  BACKBONE:
    FREEZE_AT: 2
    NAME: build_resnet_fpn_backbone
  DEVICE: cuda
  FPN:
    FUSE_TYPE: sum
    IN_FEATURES: ['res2', 'res3', 'res4', 'res5']
    NORM:
    OUT_CHANNELS: 256
  KEYPOINT_ON: False
  LOAD_PROPOSALS: False
  MASK_ON: True
  META_ARCHITECTURE: PanopticFPN
  PANOPTIC_FPN:
    COMBINE:
      ENABLED: True
      INSTANCES_CONFIDENCE_THRESH: 0.5
      OVERLAP_THRESH: 0.5
      STUFF_AREA_LIMIT: 4096
    INSTANCE_LOSS_WEIGHT: 1.0
  PIXEL_MEAN: [103.53, 116.28, 123.675]
  PIXEL_STD: [1.0, 1.0, 1.0]
  PROPOSAL_GENERATOR:
    MIN_SIZE: 0
    NAME: RPN
  RESNETS:
    DEFORM_MODULATED: False
    DEFORM_NUM_GROUPS: 1
    DEFORM_ON_PER_STAGE: [False, False, False, False]
    DEPTH: 50
    NORM: FrozenBN
```

```
    NUM_GROUPS: 1
    OUT_FEATURES: ['res2', 'res3', 'res4', 'res5']
    RES2_OUT_CHANNELS: 256
    RES5_DILATION: 1
    STEM_OUT_CHANNELS: 64
    STRIDE_IN_1X1: True
    WIDTH_PER_GROUP: 64
  RETINANET:
    BBOX_REG_WEIGHTS: (1.0, 1.0, 1.0, 1.0)
    FOCAL_LOSS_ALPHA: 0.25
    FOCAL_LOSS_GAMMA: 2.0
    IN_FEATURES: ['p3', 'p4', 'p5', 'p6', 'p7']
    IOU_LABELS: [0, -1, 1]
    IOU_THRESHOLDS: [0.4, 0.5]
    NMS_THRESH_TEST: 0.5
    NUM_CLASSES: 80
    NUM_CONVS: 4
    PRIOR_PROB: 0.01
    SCORE_THRESH_TEST: 0.05
    SMOOTH_L1_LOSS_BETA: 0.1
    TOPK_CANDIDATES_TEST: 1000
  ROI_BOX_CASCADE_HEAD:
    BBOX_REG_WEIGHTS: ((10.0, 10.0, 5.0, 5.0), (20.0, 20.0, 10.0, 10.0), (30.0, 30.0, 15.0, 15.0))
    IOUS: (0.5, 0.6, 0.7)
  ROI_BOX_HEAD:
    BBOX_REG_WEIGHTS: (10.0, 10.0, 5.0, 5.0)
    CLS_AGNOSTIC_BBOX_REG: False
    CONV_DIM: 256
    FC_DIM: 1024
    NAME: FastRCNNConvFCHead
    NORM:
    NUM_CONV: 0
    NUM_FC: 2
    POOLER_RESOLUTION: 7
    POOLER_SAMPLING_RATIO: 0
    POOLER_TYPE: ROIAlignV2
    SMOOTH_L1_BETA: 0.0
  ROI_HEADS:
    BATCH_SIZE_PER_IMAGE: 512
    IN_FEATURES: ['p2', 'p3', 'p4', 'p5']
    IOU_LABELS: [0, 1]
    IOU_THRESHOLDS: [0.5]
    NAME: StandardROIHeads
    NMS_THRESH_TEST: 0.5
    NUM_CLASSES: 80
    POSITIVE_FRACTION: 0.25
    PROPOSAL_APPEND_GT: True
    SCORE_THRESH_TEST: 0.5
  ROI_KEYPOINT_HEAD:
    CONV_DIMS: (512, 512, 512, 512, 512, 512, 512, 512)
    LOSS_WEIGHT: 1.0
    MIN_KEYPOINTS_PER_IMAGE: 1
    NAME: KRCNNConvDeconvUpsampleHead
    NORMALIZE_LOSS_BY_VISIBLE_KEYPOINTS: True
    NUM_KEYPOINTS: 17
    POOLER_RESOLUTION: 14
    POOLER_SAMPLING_RATIO: 0
    POOLER_TYPE: ROIAlignV2
  ROI_MASK_HEAD:
    CLS_AGNOSTIC_MASK: False
    CONV_DIM: 256
    NAME: MaskRCNNConvUpsampleHead
    NORM:
    NUM_CONV: 4
    POOLER_RESOLUTION: 14
    POOLER_SAMPLING_RATIO: 0
    POOLER_TYPE: ROIAlignV2
  RPN:
    BATCH_SIZE_PER_IMAGE: 256
    BBOX_REG_WEIGHTS: (1.0, 1.0, 1.0, 1.0)
    BOUNDARY_THRESH: -1
    HEAD_NAME: StandardRPNHead
    IN_FEATURES: ['p2', 'p3', 'p4', 'p5', 'p6']
    IOU_LABELS: [0, -1, 1]
    IOU_THRESHOLDS: [0.3, 0.7]
    LOSS_WEIGHT: 1.0
    NMS_THRESH: 0.7
    POSITIVE_FRACTION: 0.5
```

```
    POST_NMS_TOPK_TEST: 1000
    POST_NMS_TOPK_TRAIN: 1000
    PRE_NMS_TOPK_TEST: 1000
    PRE_NMS_TOPK_TRAIN: 2000
    SMOOTH_L1_BETA: 0.0
  SEM_SEG_HEAD:
    COMMON_STRIDE: 4
    CONVS_DIM: 128
    IGNORE_VALUE: 255
    IN_FEATURES: ['p2', 'p3', 'p4', 'p5']
    LOSS_WEIGHT: 0.5
    NAME: SemSegFPNHead
    NORM: GN
    NUM_CLASSES: 54
  WEIGHTS:
E:\Play\eye_tracking\workflow\model\model_final
_dbfeb4.pkl
OUTPUT_DIR: ./output
SEED: -1
SOLVER:
  BASE_LR: 0.02
  BIAS_LR_FACTOR: 1.0
  CHECKPOINT_PERIOD: 5000
  GAMMA: 0.1
  IMS_PER_BATCH: 16
  LR_SCHEDULER_NAME: WarmupMultiStepLR
  MAX_ITER: 90000
  MOMENTUM: 0.9
  STEPS: (60000, 80000)
  WARMUP_FACTOR: 0.001
  WARMUP_ITERS: 1000
  WARMUP_METHOD: linear
  WEIGHT_DECAY: 0.0001
  WEIGHT_DECAY_BIAS: 0.0001
  WEIGHT_DECAY_NORM: 0.0
TEST:
  AUG:
    ENABLED: False
    FLIP: True
    MAX_SIZE: 4000
    MIN_SIZES: (400, 500, 600, 700, 800, 900, 1000,
1100, 1200)
  DETECTIONS_PER_IMAGE: 100
  EVAL_PERIOD: 0
  EXPECTED_RESULTS: []
  KEYPOINT_OKS_SIGMAS: []
  PRECISE_BN:
    ENABLED: False
    NUM_ITER: 200
VERSION: 2
```

APPENDIX C LIST OF OBJECTS IN COCO PANOPTIC DATASET

Stuff classes and their stuff ids:

{"1": "banner", "2": "blanket", "3": "bridge", "4": "cardboard", "5": "counter", "6": "curtain", "7": "door-stuff", "8": "floor-wood", "9": "flower", "10": "fruit", "11": "gravel", "12": "house", "13": "light", "14": "mirror-stuff", "15": "net", "16": "pillow", "17": "platform", "18": "playingfield", "19": "railroad", "20": "river", "21": "road", "22": "roof", "23": "sand", "24": "sea", "25": "shelf", "26": "snow", "27": "stairs", "28": "tent", "29": "towel", "30": "wall-brick", "31": "wall-stone", "32": "wall-tile", "33": "wall-wood", "34": "water", "35": "window-blind", "36": "window", "37": "tree", "38": "fence", "39": "ceiling", "40": "sky", "41": "cabinet", "42": "table", "43": "floor", "44": "pavement", "45": "mountain", "46": "grass", "47": "dirt", "48": "paper", "49": "food", "50": "building", "51": "rock", "52": "wall", "53": "rug"}

Thing classes and their thing ids:

{"0": "person", "1": "bicycle", "2": "car", "3": "motorcycle", "4": "airplane", "5": "bus", "6": "train", "7": "truck", "8": "boat", "9": "traffic light", "10": "fire hydrant", "11": "stop sign", "12": "parking meter", "13": "bench", "14": "bird", "15": "cat", "16": "dog", "17": "horse", "18": "sheep", "19": "cow", "20": "elephant", "21": "bear", "22": "zebra", "23": "giraffe", "24": "backpack", "25": "umbrella", "26": "handbag", "27": "tie", "28": "suitcase", "29": "frisbee", "30": "skis", "31": "snowboard", "32": "sports ball", "33": "kite", "34": "baseball bat", "35": "baseball glove", "36": "skateboard", "37": "surfboard", "38": "tennis racket", "39": "bottle", "40": "wine glass", "41": "cup", "42": "fork", "43": "knife", "44": "spoon", "45": "bowl", "46": "banana", "47": "apple", "48": "sandwich", "49": "orange", "50": "broccoli", "51": "carrot", "52": "hot dog", "53": "pizza", "54": "donut", "55": "cake", "56": "chair", "57": "couch", "58": "potted plant", "59": "bed", "60": "dining table", "61": "toilet", "62": "tv", "63": "laptop", "64": "mouse", "65": "remote", "66": "keyboard", "67": "cell phone", "68": "microwave", "69": "oven", "70": "toaster", "71": "sink", "72": "refrigerator", "73": "book", "74": "clock", "75": "vase", "76": "scissors", "77": "teddy bear", "78": "hair drier", "79": "toothbrush"}

APPENDIX D LIST OF SAMPLE UTTERANCES TO BUILD AMAZON LEX CHATBOT IN THE CASE STUDY

Intent I – app interaction

- "zoom in", "zoom out", "I'm zooming to here", "click this", "open the tab", "I'll remove it", "close the tab", "click this button", "hit the button", "click the icon", "I'm zooming in further", "take a photo", "take a note", "I'm looking at the map", "I'm looking through the AR", "check out the map", "I can make a note here"

Intent M – movement and navigation

- "now I'm walking to", "I'm going to", "let's go to", "let's move on to", "I'm navigating to", "I'm walking on the street", "I'm crossing the road", "I'll go across", "let's go to the other side", "I'll turn right", "I now turn left", "walk straight ahead", "keep walking", "cross the street", "walking past it"

Intent T – task related

- "they are factories", "it was built on the old factory", "it is the old site", "It looks like a former factory", "it was from the textile industry", "this is not a factory", "I don't know if this was the textile industry before", "it is an original building from the textile industry", "it is a new building built on the site of old factories", "it seems related with the old industries", "this is a remanence of the original factory", "it was a remnant", "the building is still here", "the road structure is still the same", "this building was there already", "there is no remnant", "this building is new", "this building is interesting"

Intent Y – usability issues and comments

- "this is not helpful", "you should add a button", "it would be helpful if you have it", "it would be good to", "there's a problem with it", "it's difficult to use", "I don't like that", "I like it when", "it would have been nice to", "it's nice to have more information about", "I would like to have more information about", "I need more information on that", "There's something wrong with the app", "it has stopped working", "it stops working", "it is not working", "something is wrong here"

APPENDIX E USABILITY ISSUES OF GEOFARA IDENTIFIED FROM THINK-ALOUD PROTOCOLS

Type of issue	Item	Protocols (from original automated transcript)
difficulties / fault with app	VR and label	It's being chased off the screen as I turned, and so it should be in front of me. But the actual icon is being pushed off, making me think I should go this way when I know I should be going this way.
		So again, the actual augmented reality is not helpful in this case. And the map is not as helpful either.
		And if the labels were attached better to things and it could be more useful.
		Okay, so again, the method logical problem gonna actually try to look at Yeah, this is really hard to be looking at this view while walking, especially because the labels are changing so much that it's like almost making me dizzy.
		So this is an example where the augmented was helpful because I didn't have enough basemap context And so, as I'm using this, I'm realizing I'm relying on the map as much as I possibly can to navigate and to a lesser extent, my prior knowledge of the area and on Lee using the augmented when there's something that I can't determine.
	map and navigation	It looks like actually, my navigation, the direction Arrow has stopped working on the man.
		So something's wrong with the compass.
		I don't like that it keeps going to non planemetric to oblique angle.
		And because this is not an ego centric view, I don't think a plan a metric or an oblique view is appropriate. I think we should only switch to oblique if it made it forward facing meaning the map always stayed forward. I think you should stay in a planet metric view and only do oblique when you have, ah, ego centric view.
	note	Once again, you probably could tell I tried to hit the locate me button when I meant to open the notes, and so something that would be a clear you I would be good.
		I don't like that My note comes up every time because you actually have to delete it.
		Okay, So that note I did something wrong with that note.
		Okay, once again, hitting back is a weird way to save a comment.
	others	It's getting windy out here, making a little more challenging to use this application.

need for additional information	base map	This is a case where a satellite image would be really helpful for the base map.
		It would have been helpful to rather than have a point on the map to have the actual footprint of the thing because sometimes the shape of the building um it's helpful to connect what I'm seeing with what's on the map.
		And so it doesn't give me a sense of the Kwantum is actually the things that I should be looking at here.
		And really, one thing I found is to have just points could be problematic.
		I think maybe if it's a layer you can turn on and off.
	additional POI information	So it's a little confusing that I didn't get any context about that because cruises a restaurant today, but I don't know what this actually was.
		Maybe including this banner would help Thio make this landmark more salient in the landscape where you could actually confirm where you're going. If I knew what I was looking for, a banner that would've been helpful, I would've looked up rather than at my phone.
		So something that related this particular building to factories in the past would be helpful.
		(would) be interesting if these maps were annotated when you come to the specific site with kind of, ah, where what you're looking at.
		Okay , It would have been helpful to have this Praxis sign up and somewhere in here to confirm.
		So an image not from that far side but from this side would be helpful.
		And so, if you need to be able to see this first in some way, and then look at the modern , the modern buildings and how they've been either Tauron down on or re facade ID because the more.
		It would have been good to actually annotate these maps with the boundary of the wall so I can confirm exactly where I am.
		And so as I mentioned back there, a new photograph should be taken from that point s so that you have the correct photo from where you're standing.
		And then when you pull up the images, it would be good to actually mark on these images. Yeah, And do it every time consistently to make it so that there's a cognitive association between the historic image.

		Just because it doesn't tell me it doesn't give me any context about how it relates to textiles.
		So some clarification about the history would be interesting, because right now I feel like I'm walking it pretty far away to see a tunnel.
		I wonder if there's any information about the architecture that could be given here.
		I'd be interested to know how it relates and again and be helpful , actually, have it on the map maps here to be able to relate all the maps together.
	restriction	But one thing of having some sort of understanding of where I can go in where I shouldn't go on the map would be helpful.
		So it's another situation would be helpful to know where you can and can't go And what time of day.
		The tour is also taking me in a location where there's not a sidewalk , which sometimes avoiding locations that don't have a sidewalk, might be a good way to make sure that you don't.
recommendation on functionality		So one thing that would be really neat is if the app signaled which locations you've already been to. Both that you've already clicked information, but also were sort of geo fence was in a particular area so that you know that you've actually physically been there because that would help me determine which points I haven't been to yet.
		So on ordering might actually be helpful, recommended ordering might actually be helpful.
		And so the fact that I can't see the age a store from here, if you better maybe if I was being led on the ground with a direction, an arrow to it.