

Designing & Implementing A Geospatial Mobile Application

Alfred Hirschfeld, Christoffer Karlsson

DEPARTMENT OF DESIGN SCIENCES & PHYSICAL GEOGRAPHY AND
ECOSYSTEM SCIENCE | FACULTY OF ENGINEERING LTH
LUND UNIVERSITY 2022

THESIS IN GEOGRAPHICAL INFORMATION TECHNOLOGY NR 32

THESIS IN INTERACTION DESIGN

heig-**vd**

HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

[www.heig-**vd**.ch](http://www.heig-vd.ch)



Designing & Implementing A Geospatial Mobile Application

Alfred Hirschfeld
al0015hi-s@student.lu.se

Christoffer Karlsson
ch1322ka-s@student.lu.se

February 24, 2022

Master's thesis work carried out at Haute École D'Ingénierie et de Gestion
du Canton de Vaud (HEIG-VD/HES-SO).

Supervisors:

Jens Ingensand, Institute of Territorial Engineering, HEIG-VD
Lars Harrie, Department of Physical Geography and Ecosystem Science, LU
Olivier Ertz, Institute of Media Engineering, HEIG-VD
Joakim Eriksson, Department of Design Sciences, LU

Examiners:

Günter Alce, Department of Design Sciences, LU
Ali Mansourian, Department of Physical Geography and Ecosystem Science, LU

Designing & Implementing A Geospatial Application

Copyright © 2022 Alfred Hirschfeld and Christoffer Karlsson

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Department of Physical Geography and Ecosystem Science
Faculty of Science LTH, Lund University
Sölvegatan 12, SE-223 62 Lund, Sweden

Subjects: Interaction Design (MAMM01), Geographical Information System (EXTM05)
Division: Media Engineering Institute of HEIG-VD
Supervisors: Jens Ingensand, Lars Harrie, Olivier Ertz, Joakim Eriksson
Examiners: Ali Mansourian, Günter Alce

Abstract

Rephotography is the act of recapturing a historic photo from the same viewpoint and is a tool for studying change. This thesis explores the development, design and evaluation of a mobile application for rephotography of Smapshot photos. Rephotography would have to be based on sensors in smartphones and previous research has demonstrated a lack of accuracy when computing the orientation with these sensors. We implemented two techniques for rephotography: orientation angles of smartphones and transparency of Smapshot photos. Our findings shows that from a user perspective transparency is preferred, however orientation angles result in a more accurate re-creation of the historic photo.

Keywords: Interaction design, geographical data, rephotography, mobile application development

Sammanfattning

Omfotografering är handlingen att återfånga ett historiskt foto från samma synvinkel och är ett verktyg för att studera förändring. Detta examensarbete utforskar utvecklingen, designen och utvärderingen av en mobilapplikation för omfotografering av Snapshot-foton. Omfotografering i en mobilapplikation skulle behöva baseras på sensorerna i mobilen och tidigare forskning har visat på bristande noggrannhet när man beräknar orienteringen med dessa sensorer. Vi implementerade två tekniker för omfotografering: orienteringsvinklar för smartphones och transparens för Snapshot-foton. Våra resultat visar att ur ett användarperspektiv föredras transparens, men orienteringsvinklar resulterar i ett mer exakt återskapande av det historiska fotot.

Nyckelord: Interaktionsdesign, geografisk data, omfotografering, apputveckling

Acknowledgements

We would like to thank our supervisors from HEIG-VD: Jens Ingensand and Olivier Ertz for having us, engaging us in a fun project and for making us feel welcome at HEIG-VD. We would also like to thank Lars Harrie for introducing us to the opportunity of doing our master's thesis together in Switzerland. Thank you to Lars Harrie and Joakim Eriksson for supporting us through the process and guiding us towards the completion of our thesis project. Lastly, we would like to thank Stéphane Lecorney for always being helpful.

Yverdon-les-Bains, January 2022

Alfred Hirschfeld & Christoffer Karlsson

Abbreviations

AGNSS - Assisted Global Navigation Satellite System

API - Application Programming Interface

DGNSS - Differential Global Navigation Satellite System

FOV - Field of View

GCP - Ground Control Points

GIS - Geographic Information System

GNSS - Global Navigation Satellite System

RMS - Root Mean Square

SD - Standard Deviation

SDLC - Software Development Life Cycle

SMA - Snapshot Mobile Application

SUS - System Usability Scale

UCD - User-Centered Design

VGI - Volunteered Geographic Information

Contents

1	Introduction	9
1.1	Background	9
1.2	Purpose	11
1.3	Research questions	12
1.3.1	Research questions related to geographic information	12
1.3.2	Research questions related to design	13
1.4	Limitations and assumptions	13
1.4.1	Test-users	14
1.5	Thesis structure	14
2	Snapshot	15
2.1	What is Snapshot?	15
2.2	Mobile application	16
3	Geospatial Theory	17
3.1	Spatial orientation	17
3.1.1	Coordinate systems	17
3.1.2	Map projections	17
3.1.3	Smartphone sensors	19
3.1.4	Sensor coordinate system	19
3.1.5	Accelerometer and geomagnetic field sensor	20
3.1.6	GNSS sensor and barometer	20
3.1.7	Altitude	21
3.1.8	Sensor coordinates to device coordinates	22
3.1.9	Computing the device orientation with Android OS	23
3.2	Native applications and web applications	24
3.3	Focal length and principal point	25
3.4	Georeferencing in Snapshot	26
3.4.1	Monoplotting	28
3.4.2	Uncertainty estimation and validation in Snapshot	30

3.5	Rephotography	31
3.6	Data quality	31
3.6.1	Terms related to statistical data assessment	32
3.7	Rephotography data quality in smartphones	32
3.7.1	Location and orientation	32
3.7.2	Pitch and roll	33
4	Design Theory	35
4.1	Iterative design process	35
4.2	Brain- & bodystorming	36
4.3	User-centered design	36
4.3.1	Personas	37
4.3.2	Prototyping	38
4.4	Mobile interaction design	38
4.4.1	Design principles	39
4.5	Gamification	40
4.6	Design practices for maps	41
4.7	Usability testing	41
4.7.1	Think aloud	41
4.7.2	Remote usability testing	42
4.7.3	Number of participants	42
4.7.4	System Usability Scale	43
5	Design Process	45
5.1	First iteration	45
5.1.1	Brain- & bodystorming	46
5.1.2	Storyboarding	47
5.1.3	Survey	47
5.1.4	Defining end-users	48
5.1.5	Personas	49
5.1.6	Scenarios	50
5.1.7	Low fidelity prototype	51
5.2	Second iteration	53
5.2.1	High fidelity prototype	53
5.2.2	First usability testing	55
5.2.3	Mobile development	58
5.3	Third iteration	59
5.3.1	Partly implemented prototype	59
5.3.2	Second usability testing	62
6	Development implementation	67
6.1	Technical Background	67
6.1.1	Selection of photos	68
6.1.2	Application programming interface	69
6.1.3	Dependencies	69
6.1.4	Github repository	70
6.1.5	Agile Software Development	70

6.2	Development process	70
6.3	Implementation	71
6.3.1	Portrait vs. landscape mode	71
6.3.2	Compensating for magnetic declination	71
6.3.3	Normalisation of orientation angles	72
6.3.4	User instruction	72
6.3.5	Comparing the orientation angles	74
7	Rephotography Data Quality Assessment	75
7.1	Test 1	75
7.1.1	Historic photo results	76
7.1.2	Test photo results	79
7.2	Test 2	80
7.3	Test 3	81
8	Discussion	85
8.1	Mobile application	85
8.1.1	Techniques for retaking photos	86
8.1.2	Gamification	87
8.2	Design process	88
8.3	Rephotography data quality assessment	90
8.4	Future work	92
9	Conclusion	93
9.1	Geographic information conclusions	93
9.2	Design conclusions	94
	References	95
	Appendix A Survey	103
	Appendix B Usability Test Plan	105
B.1	Second iteration	105
B.2	Third iteration	106
	Appendix C Rephotography data quality tests	109

Chapter 1

Introduction

1.1 Background

Volunteered Geographic Information (VGI) is data that has been created, assembled and disseminated by volunteers. The term VGI was first introduced in 2007 [1]. VGI engages private citizens in the creation of geographic information, which is something that historically have been reserved for official agencies. One of the most well-known VGI-projects is OpenStreetMap which is a free, editable map of the whole world. Other examples include a service that enables users to gather and share disaster information in real time [2] or a mobile application that supports the monitoring of the endangered cheetahs, by citizens [3]. Another example of a VGI-project is Smapshot [4].

Smapshot is a web-based platform with the goal of creating a virtual globe of history. Historical photographs from various collections can be aligned with the virtual globe by having volunteers georeference the historical photographs. Georeferencing means that you define the location and direction from which a photo was taken. Smapshot is an open source project and has been in development since 2017 by HEIG-VD in Switzerland, containing 200,000 photos. Open source means that code is publicly available to modify and share by anyone, which means that developers can build their own tools and applications on already existing and freely-available work [5]. The photos of Smapshot are collections from various archives, foundations and institutions in Switzerland [6].



Figure 1.1: Example of a historic photo aligned with a virtual globe, from Smapshot's website [4]

When georeferencing historical photographs in Smapshot, the location from which a photo was taken is defined. The direction of the camera is also computed. This direction, or orientation of the camera, consists of three angles. These parameters can be used for rephotography. Rephotography is the act of recapturing the same scene from an existing photograph and from the same viewpoint. Often, the latter photograph is older. Creating a rephotograph that is faithful to the original photograph without reproducing the location and direction of the camera, and instead perform this act of exactly knowing a photograph's viewpoint "by eye", is challenging [7]. Example of a recent rephotography project is the mobile application Ajapaik.

In mobile devices there exist computational techniques for rephotography that does not rely on the human eye [8]. Other studies has demonstrated the ability of smart phone applications used to guide users to their locations of photos to recapture [9]. However, these studies do not discuss the design aspect of such work.

The goal for this master thesis is to use generated parameters of photographs from Smapshot for rephotography in a mobile application and create a design that is influenced by end-users. Rephotography can be used for visualizing the progress of time, making it a tool for studying changes in nature and cities. A well matched photograph and its rephotograph is a clear way of illustrating which elements in the scene has changed across time, and which elements has been preserved. For example, rephotography is used to document glacier melting as a consequence of global warming [10] and to monitor geological erosion and change [11].

1.2 Purpose

This master thesis deals with both geographic information- and design-related issues. The purpose, goals and research questions of the thesis can therefore be divided into two areas: geographic information and design.

Goals related to geographic information

Through the georeferencing process performed by Smapshot, the camera position, direction and field of view of the historical photographs are derived. The purpose is to research the possibility of using these parameters in order to retake the historical photographs. This is researched through the development of a mobile application. The application will instruct the user how to position him- or herself, and how to angle the phone to be able to retake the historical photograph.

Furthermore, an application such as the one suggested, especially with the availability and the public's generally high knowledge of the of mobile devices, could increase interest in the Smapshot project. This would lead to more contributors of this VGI-project, allowing for more photographs to be georeferenced, which in turn would allow for more images to be used in other ways related to geographic information.

On the one hand there is a need to research the accuracy related to the sensors of a mobile device, because they are used to calculate the spatial position and orientation of the device. On the other hand there are the errors of the location and orientation parameters from the historical photographs, which are computed by the georeferencing process. Testing the prototype application by retaking photographs and then comparing the orientation parameters with the historical photos is also needed.

Goals related to design

The aim of this thesis is to design and develop a mobile application prototype with focus on rephotography. This is where a user-centered design (UCD) process becomes important: a process which underlines the importance of having end-users influence the development and design of a product [12].

Since the mobile application prototype will allow users to retake historical photographs, the design-related work focus on creating a design, influenced by the intended end-users. The main focus is to create a user-friendly design as well as accomplish product usefulness [13]. A UCD process can vary depending on project and in this case, UCD includes research about who the intended user group might be in order to make correct design decisions and mitigate the risk of spending unnecessary time on developing and designing features no one needs [14].

A part of the UCD process is to assess the design by having test-users evaluate the current design and to collect performance- as well as preference data, used for further improvements in the next iterations of the UCD process. Another aspect related to design is to research gamification and its influence towards the end-user with the goal of determining if it engages the user.

1.3 Research questions

The goal is to develop a Smapshot mobile application prototype with focus on functionality for recapturing photographs. This goal can be broken down into two fields: geographic information and design.

1.3.1 Research questions related to geographic information

The goal is to design, implement and evaluate a method for the user to retake a historical photograph in a Smapshot mobile application prototype. This can be broken down into:

- The development of a prototype application that extracts and saves the spatial orientation of a mobile device at the moment of capturing a photograph.
- Give instructions to the user on how to recapture the photograph.
- The comparison of the device's orientation to that of a historical photograph.
- Assess the uncertainties of orientation data from the georeferencing of Smapshot, and the rephotography process.

In order to fulfill the goals presented above, the following research questions are addressed:

1. *What kind of uncertainties can be expected when rephotographing a Smapshot photo (with known orientation) using a smartphone?*
2. *How can a Smapshot mobile application be implemented for rephotography?*
3. *What problems have to be addressed in relation to the computation of the orientation angles of the device, and the comparison to the orientation angles of photos from Smapshot?*

1.3.2 Research questions related to design

The goals of designing a mobile application prototype for Smapshot consists of:

- Carrying out a user-centered design process.
- Including design practices for gamification.

In order to fulfill the goals, following research questions need to be addressed:

1. *To what extent is there an interest of using a Smapshot mobile application?*
2. *Who are the potential end-users?*
3. *What are the challenges in its usage?*
4. *How to keep interest of usage?*

1.4 Limitations and assumptions

There are limitations in the study to mention that affect the outcome of the thesis result.

Time frame

The time-frame is one aspect, where the work of this thesis was conducted in 21 weeks, starting from Monday 23rd of August 2021 to Friday 28th of January 2022. In addition, the number of iterations in the UCD process was limited to fit the time scope to three iterations in total.

The scope of this work focus on the functionality of recapturing photographs by comparing orientation angles of cameras. Several more features could be included such as integrating a photo library from Smapshot, navigating the user to the viewpoints of photos and storing rephotographed photos in a database. Once again, due to lack of time, implementing this was not feasible.

Technical limitation

Working in Android Studio is a technical limitation since the mobile application runs solely on the Android operating system and not on iOS. There is also a limitation in the number of devices that the application was tested on, currently running on two Android smartphones, one Xiaomi Mi 9T and one Motorola G8. This limitation means that the application was tested and made fit for the hardware on these two devices.

1.4.1 Test-users

The covid-19 pandemic affected the availability of recruiting test-users for evaluating the design work of the mobile application. Another limitation was the language barriers. Since a few potential test subjects did not speak English nor Swedish they had to be excluded as candidates for the usability tests.

1.5 Thesis structure

The first part of this master thesis is a short introduction to Smapshot. It explains the service that Smapshot provides, and what the work of this contributes to the Smapshot project. The following chapters presents literature which deals with theory regarding spatial data, how the device orientation is calculated, relevant work in the field of rephotography, as well as studies related to the accuracy of the sensor used to gather spatial data in mobile devices. It includes a basic introduction to coordinate systems, projections and smartphone sensors. The studies related to rephotography illustrates the use of these kinds of applications. In addition it highlights common difficulties and solutions to problems related to rephotography. This part also includes theory related to interaction design such as design theory, design principles, usability testing and gamification.

The second part of this master thesis, which starts at chapter five, deals with the practical assets of developing the functionality of the mobile applications, the design process as well as the practical part of evaluating the application. Further, the results from the design process, the tests, and the final prototype application is presented and discussed.

Chapter 2

SmAPSHOT

2.1 What is SmAPSHOT?

Historical images have a high cultural, artistic and scientific value. However, due to the poor labeling of the images, as well as the absence of georeference, database searches are inefficient, and geographic information can not be extracted from the images. In the context of landscape imagery, this means that the detailed state of landscape, recorded by the imagery, cannot be inserted in a GIS, since it is not related to a geographic coordinate system. This prevents exploitation of the images within Geographic Information Science. However, the information required for labeling images of landscapes (for instance names for places and land covers) are generally stored in spatial databases. This is why a system dedicated to linking these images with the spatial data already stored, is of value [6].

Web-based 3D solutions are increasing and so does the public's familiarity of 3D software and virtual globes. SmAPSHOT aims to make use of this familiarity to offer a tool that allows for volunteers to georeference images. In order to relate a photograph with geographic data the position of the camera that took the photograph, the direction of which the camera was facing, and the field of view must all be calculated. A project with the purpose of allowing private citizens to voluntarily georeference images is an efficient way of georeferencing image-collections, and consequently allowing for a much broader use of historic images, like rephotography. Georeferenced images enables scientists to extract geographic data from the images using photogrammetric methods. This extracted data could then be used to calculate the volume of a glacier at different times, or to recreate three-dimensional models of destroyed buildings. The goal of SmAPSHOT is not only to make use of volunteers in order to compute the location and orientation of historic photographs, but also to visualize the photo as a 3D object in a virtual globe [6], like illustrated in Figure 1.1.

2.2 Mobile application

SmAPSHOT is experimenting with the idea of expanding its work to mobile platforms, allowing users to recapture the georeferenced, historical photographs. The purpose of this is to compare the past with the present. It is also a way for SmAPSHOT to attract more volunteers. The description of the master thesis proposal stated:

"The goal is to implement a mobile application for the visualisation of georeferenced photos. The mobile application can show the past state of the landscape (historical images textured on the landscape model) or the location of the user in the pictures."

Chapter 3

Geospatial Theory

3.1 Spatial orientation

3.1.1 Coordinate systems

Spatial data contains numerical information that specifies its location on earth. This numerical information is based on a coordinate system, which provides a geographic reference frame for the data. This allows the data to be aligned relative to other spatial data. The data can be defined horizontally, defining the position of the data across the surface, and vertically, defining the locations height or depth. There are several transformations that can be used to convert data between the different coordinate systems [15].

3.1.2 Map projections

When displaying spatial data on a flat surface, like a piece of paper or the screen of a mobile phone, a map projection is required. The surface of the earth is curved, and there is no way of displaying it on a flat surface without some distortions. Mathematical calculations can be used to convert the coordinate system to a flat surface, but it is not done without some distortions, like distortions to area or shape. Different ways of converting coordinate systems, bring along different qualities, some preserve the correct shape, while others preserve the area. Consequently, different map projections are suitable for different tasks. The shape of the Earth is approximated when it comes to projections. The shape can be approximated as a sphere or as an ellipsoid [15]. Different map projections use different projection surfaces. The surfaces need to be able to be unfolded into a plane. One example of such a projection surface is a cylinder. A cylindrical map projection, where the graticule of a coordinate system is projected onto a cylindrical projection surface is, illustrated in Figure 3.1.

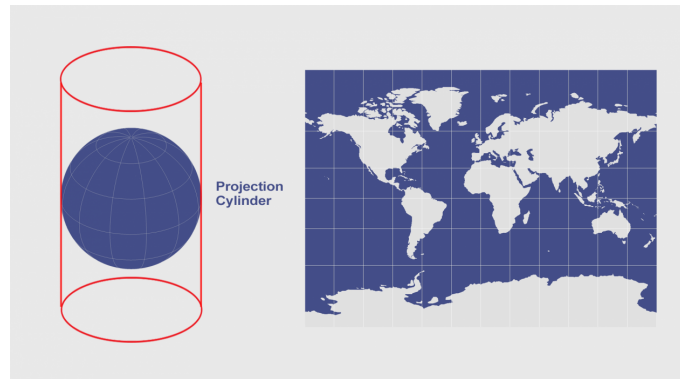


Figure 3.1: The graticule of a coordinate system is projected onto a cylindrical projection surface in a cylindrical projection [16], © 2022 GIS Geography

The virtual globe of Snapshot is based on Cesium, an open developer platform for 3D software applications. Cesium is based WGS84 coordinate system and uses a Web Mercator projection. [17]. The Web Mercator projection is the standard projection for Web mapping applications. It is a variant of the Mercator projection, which uses a cylindrical projection surface. The formulas for Web Mercator projection is based on a spherical form, but the geographical coordinates are in WGS84, which are based on an ellipsoidal form. This causes the Web Mercator projection to be slightly non-conformal. A spherical Mercator projection results in areas expanding with the distance from the equator. Meaning areas in the polar regions are magnified. This is the case with Greenland, visualised in Figure 3.2 [18].



Figure 3.2: The distortion of Greenland as a consequence of the Mercator Projection [19], Copyright © 2021 Visual Capitalist

3.1.3 Smartphone sensors

Android smartphones have sensors that enables measuring of the motion and of the orientation of the device. This provides an opportunity for smartphones to be used for research purposes. Most people carry smartphones and in the context of VGI, this is indeed an advantage. This offers the possibility for volunteers to gather data, with a measurement device that they are used to, and have access to. However, the accuracy of this sensor data is limited, and it can differ between smartphones. Deviations in pitch and roll, which are orientation angles based on data from sensors, differs between smartphone models. The mean inaccuracies are between 2.1° and 6.6° [20].

3.1.4 Sensor coordinate system

To express the location of a point in a three dimensional space one can use a Cartesian coordinate system. The location of the point is then expressed through three numbers, the distance to three mutually perpendicular planes. Sensors in the Android operating system use a coordinate system that is defined relative to the screen of the device, when the device is held in its default orientation, which is in portrait mode (held vertically). If the device is held in the default orientation, the x axis is horizontal and points to the right, the y axis is vertical and points up to the sky, and the z axis points toward the outside of the screen face, along the lens of the camera. This is illustrated in Figure 3.3. If it is not addressed, the coordinate system will not change as you move the device from its default orientation. Meaning that the axes are not swapped as you move the device from portrait mode to landscape mode (held horizontally) [21].

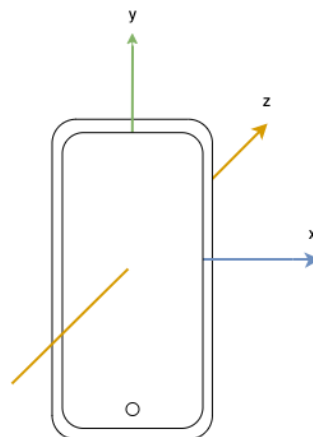


Figure 3.3: Sensor Coordinate System

3.1.5 Accelerometer and geomagnetic field sensor

In Android devices there are base sensors and composite sensors. The base sensors rely data from a single physical sensor. The composite sensors generate data from a composite of several sensors [22].

The accelerometer and the geomagnetic field sensors (compass sensor) are the two sensor which are used when calculating the orientation of a device. They are both base sensors. The accelerometer measures linear acceleration along the axes of Figure 3.3. The geomagnetic field sensor estimates the magnetic field at a given point on Earth for each of the three coordinate axes. This can be used to compute the direction in relation to the magnetic north. [23]

True north is the direction along the surface of the Earth towards the geographic north pole. True north differs between coordinate systems, with WGS84 being used to define true north in Android [24]. True north is different from the magnetic north, which is the direction a compass points to the magnetic north pole, due to that the earth's magnetic field points vertically downward at this point. The magnetic declination is the difference between the magnetic north and True north [25]. Adding the magnetic declination to the magnetic north result in True North. Whether or not an application that make use of the compass sensor has the ability to apply the magnetic declination to the readings to simulate directions based on true north depends on the applications developer. There are models available that represents the magnetic field. These can be used for providing the magnetic declination [26].

In the context of computing the orientation of the device, the accelerometer is used solely for the purpose of detecting which way is up and which way is down. The accelerometer returns the acceleration along the three axes. The values are in SI unit m/s^2 . This acceleration is derived by measuring forces applied to the sensor, with gravity being one such force. With the gravity constantly influencing the sensor, the accelerometer reads an acceleration of around $9.81 m/s^2$ if it is lying still on a table [23].

The geomagnetic field sensor is used to determine cardinal direction, like north, by measuring the geomagnetic field strength along the three axes. Theoretically, when you stand still and change the orientation of the device, the data from the geomagnetic field sensor does not change, since the earth's magnetic field strength only changes if you move to another location. The data from the accelerometer however, would change because the linear acceleration of the device would change. The accelerometer tells the orientation code how to translate the results from the geomagnetic field sensor, by detecting which way in relation to the device is up. The measurements from the geomagnetic field sensor are values in micro-Tesla (uT), which is a unit for measuring the magnetic field [23].

3.1.6 GNSS sensor and barometer

Other sensors that may be used to measure spatial orientation is the barometer and a GNSS sensor. The barometer measures air pressure, which can be used when calculating the altitude. The GNSS sensor receives signals from satellites. If the time it takes to receive the signal, and the signal speed is known, it is possible to calculate the distance from the phone

to the satellites. Further, the position and speed of each satellite is known. When the sensor is able to receive signals from four satellites, it is possible to determine the position of the device on earth [23], see Figure 3.4. Most mobile devices work with an integration of several kinds of techniques for obtaining the positional location of the device. The standard technique is GNSS with WiFi information, where one such technique is assisted GNSS (A-GNSS) [27].

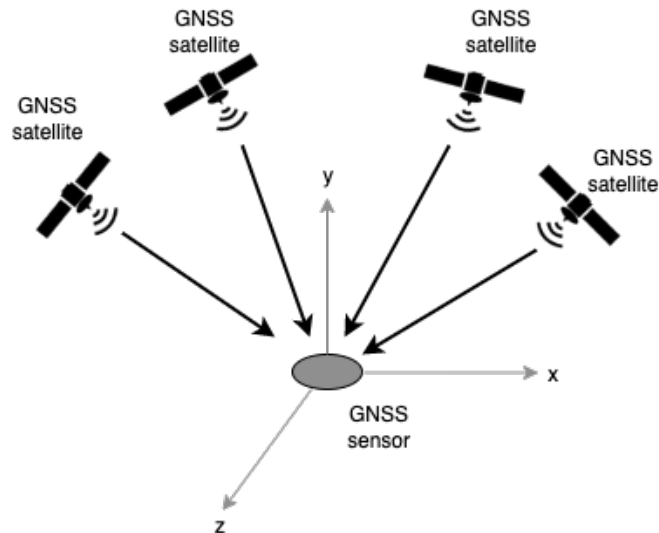


Figure 3.4: The working principal of GNSS

If a device uses assisted GNSS, it is used in combination with GNSS. Cellular towers continuously receive and store GNSS signals. The device can then use the cellular network to receive information regarding the position of satellites. This means that the device can calculate its position even when the satellite signal is weak. It can also be used to improve the accuracy of the computed location [28].

Differential GNSS (DGNSS) is a term for enhancements to GNSS location accuracy, by using several GNSS receivers [29]. A network of fixed ground based reference stations, with known location, can be used to apply corrections in real time to the location computed by the GNSS satellite system. One such DGNSS solution to GNSS measurements of mobile devices is proposed in a study, where the position accuracy was found to improve by 30-40 % [30].

3.1.7 Altitude

The shape of the earth can be approximated as an ellipsoid. The altitude can be computed as height above the reference ellipsoid. But it can also be defined as the height above the geoid. The geoid is a surface which is determined by the earth's gravity and approximated by the mean sea level[31]. When using GNSS to compute the altitude of Android devices, the altitude is given in meters above the WGS 84 reference ellipsoid. Corrections to the altitude can be applied to calculate the geoid height instead [32].

3.1.8 Sensor coordinates to device coordinates

The orientation of the device can be defined by as a combination of three angular quantities: the azimuth, the tilt and the roll as illustrated in Figure 3.5. Each orientation angle is a rotation around an axis.

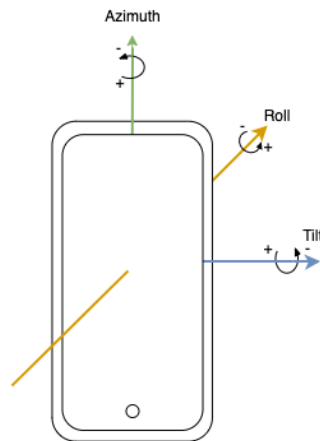


Figure 3.5: Orientation angles

The accelerometer and the geomagnetic field sensor are used for computing the orientation of the device. The data from the accelerometer and the geomagnetic field sensor can be seen as vectors in three dimensional space. We can call them a and m . Vectors can be represented by a 3-tuples (v_1, v_2, v_3) with respect to some basis (x, y, z) . This could be represented like this: $(v_1x + v_2y + v_3z)$, where (v_1, v_2, v_3) are the coordinates of some vector (for example a or m), with respect to the basis (x, y, z) . For the sensor data in Android devices, the basis is based on the coordinate system illustrated in Figure 3.3 (x, y, z) , which is relative only to the device itself. The data from the sensor changes when the orientation of the devices changes, because this changes the basis (x, y, z) . The basis (x, y, z) can be thought about and expressed as a function of time $(x(t), y(t), z(t))$, a moving coordinate system [21].

The data from the accelerometer and the geomagnetic field sensor needs to be expressed in relation to earth's frame of reference. Meaning that the coordinates of a and m needs to be expressed with respect to the basis of the world (E, N, U) (Figure 3.6) instead of the basis of the device (x, y, z) . In Snapshot, the orientation angles are in relation to True North. This means that the basis of the world (E, N, U) needs to be in relation to True North. However, the Android Operating system computes the orientation angles in relation to the magnetic North Pole [21]. Therefore, the magnetic declination needs to be compensated for when implementing the application [21].

This basis of the world (E, N, U) is defined as illustrated in Figure 3.6. The N-axis is tangential to the ground at the device's location and point towards True North. The N is known with the help of the geomagnetic field sensor and information of the magnetic declination. The U-axis points towards the sky, the opposite direction of the plumb line, and is known with the help of the accelerometer which measures the applied force of gravity. The E-axis

is the vector product of NU . The E-axis is tangential to the ground at the device's current location and points east [21]. From the Android Operating system the azimuth is the angle between the current compass reading and N. In order to express it in relation to True North instead, the magnetic declination is added to the azimuth [23].

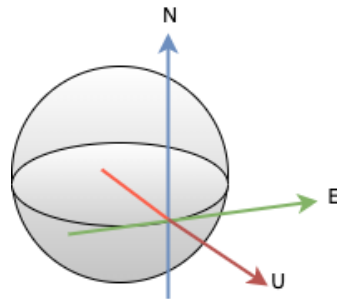


Figure 3.6: Device Coordinate System

In order to remap the data provided by the sensors in Android smartphones from the basis of (x, y, z) , to the basis of (E, N, U) , the use of a rotation matrix is required. A rotation matrix is a transformation matrix, that is a mathematical way of describing rotation in a Euclidean space[33]. The rotation matrix is used to express a point or a vector in the coordinate system relative to the screen of the device (Figure 3.3) as a point or a vector in the coordinate system relative to True North (Figure 3.6). For example, the vector from the geomagnetic field sensor m is expressed relative to the world's coordinate system (E, N, U) by the product of the rotation matrix and the values return from the sensor [21].

3.1.9 Computing the device orientation with Android OS

The accelerometer and the geomagnetic field sensor return values for each axes. They are stored in arrays. The operation system uses the data from these sensors to compute the rotation matrix R . The orientation angles (azimuth, tilt and roll) are embedded in R . The Android operating system calculates the rotation matrix R [21] [34]:

$$\begin{bmatrix} E_x & E_y & E_z \\ N_x & N_y & N_z \\ U_x & U_y & U_z \end{bmatrix}$$

Where x, y and z are the axes illustrated in Figure 3.3 and E, N and U are vectors where:

- $E = (E_x, E_y, E_z) =$ a unit vector which points East.
- $N = (N_x, N_y, N_z) =$ a unit vector which points North.
- $U = (U_x, U_y, U_z) =$ a vector which points the opposite direction of the plumb line, which roughly equals a unit vector pointing away from the centre of the earth.

After the calculation of the rotation matrix \mathbf{R} , the spatial orientation, meaning the angles azimuth, tilt and roll, can be derived. The rotation matrix is applied to project the coordinate system (x, y, z) (Figure 3.3) to coordinate system of (E, N, U) (Figure 3.6). The origin for both coordinate systems will be same, because the axes are only rotated. The three orientation angles are the rotations around the axes of the projected coordinate system (Figure 3.6). They relate to the device as illustrated in Figure 3.5.

- The azimuth (ϕ) is the rotation around the U-axis as defined in Figure 3.6. Facing north would result in 0, facing south would result in π .
- The tilt (θ), sometimes known as the pitch, is the rotation around the E-axis as defined in Figure 3.6.
- The roll (ψ) is the rotation around the N-axis as defined in figure 3.6.

The relationship between the rotation matrix \mathbf{R} that transforms a point or a vector from the coordinate system relative to the device (Figure 3.3), to the coordinate system relative to True North (Figure 3.6), and the azimuth (ϕ), the tilt (θ) and the roll (ψ) is given by [34]:

$\mathbf{R} =$

$$\begin{bmatrix} \cos \phi \cos \psi - \sin \phi \sin \psi \sin \theta & \sin \phi \cos \theta & \cos \phi \sin \psi + \sin \phi \cos \psi \sin \theta \\ -\sin \phi \cos \psi - \cos \phi \sin \psi \sin \theta & \cos \phi \cos \theta & -\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta \\ -\sin \psi \cos \theta & -\sin \theta & \cos \psi \cos \theta \end{bmatrix}$$

3.2 Native applications and web applications

There is a difference in how to gather a device's orientation from sensors between using native apps and web apps. Native apps run on top of the device's operating system, using compiled code (e.g in Java). Native applications are able to collect data from hardware sensors in the smartphone directly. Web apps, run within a web browser, like Google Chrome. Web apps use web APIs (Application Programming Interfaces) through Javascript code. Web apps are unable to gather data directly from hardware sensors, like native apps, for security reasons. Most native applications however, use what is called "software sensors", which is estimations of position and orientation by combining the readings of the hardware sensors. Then there are hybrid applications that are native pieces of code that can access device's sensors and allow for web development. In most cases however, the accuracy of measurements from web browsers and native applications are much the same. [20].

3.3 Focal length and principal point

The focal length of a camera is the distance between the optical centre of the lens and the camera's sensor. The focal length is illustrated in Figure 3.7. The focal length affects the FOV (field of view). A shorter focal length, results in a wider field of view. The primary camera of mobile devices typically has a focal length of 24mm to 27mm, and a standard focal length for a camera is usually between 40mm and 60mm. The principal point is the point on the image plane onto which the perspective center is projected. The focal length is measured from this point. These calibration effects the outcome of the image [35].

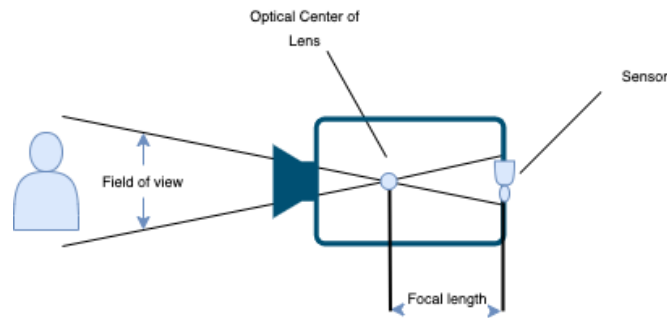


Figure 3.7: Illustrating how the sensor, the optical center of the lens and focal length affects the field of view

3.4 Georeferencing in Smapshot

In order to georeference historic photographs, the volunteers identify locations that are the same in the photograph and a 3D visual representation of reality. These corresponding points are called Ground Control Points (GCPs) and the visual representation is a virtual globe.

The virtual globe is created using the open source virtual globe API (www.cesium.com). The virtual globe renders data (digital elevation model (DEM) and aerial images) that has been provided by the Swiss Federal Office of Topography, Swisstopo. The resolution of the aerial images range from 25 to 50 cm and the DEM has a resolution of 1 m. Smapshot uses the coordinate system WGS84. The web-based virtual globe does not only provide for a way to identify GCPs but it also used for visualization of the georeferenced photographs. After the photograph is georeferenced it is inserted in the virtual globe at the computed location of the photograph [6].

To 3D georeference a photograph the 3D location, the 3D orientation and camera parameters like the field of view, are required to be computed. These parameters are calculated using the GCPs. The volunteers picks a minimum of six GCPs in both the historical photograph, which is in 2D, and the textured 3D model of the landscape, this is illustrated in Figure 3.8 [6]. Examples of locations that could be identified in the photograph and in the virtual globe are mountain tops, churches or roads, things that have remained mostly unchanged from when the photograph was taken.

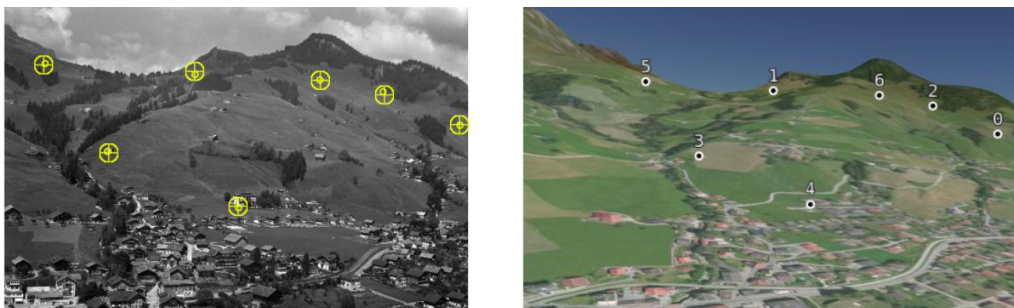


Figure 3.8: Ground Control Points, in a historical 2D photograph to the left and corresponding points in the 3D virtual globe [6]

In order to identify the GCPs the virtual globe must be position in roughly the same location as the photograph. Since a photo often has a location name associated with it, this can be used for positioning the virtual globe. In Smapshot, the first step in navigating the virtual globe to reach the location of the picture is to navigate a 2D map where photographs are marked on the map according to their location name. Then the volunteer approximates the location of the camera position on a 2D map, see Figure 3.9. The next step is that the volunteer specifies the direction of the camera (Figure 3.10). The location and direction are then used to position the virtual globe in such a way that the user will see a scene in the 2D globe similar to the 2D historical photograph. The user can then identify the GCPs (Figure 3.11)[6]. Another example of identified GCPs can be seen in Figure 3.8.

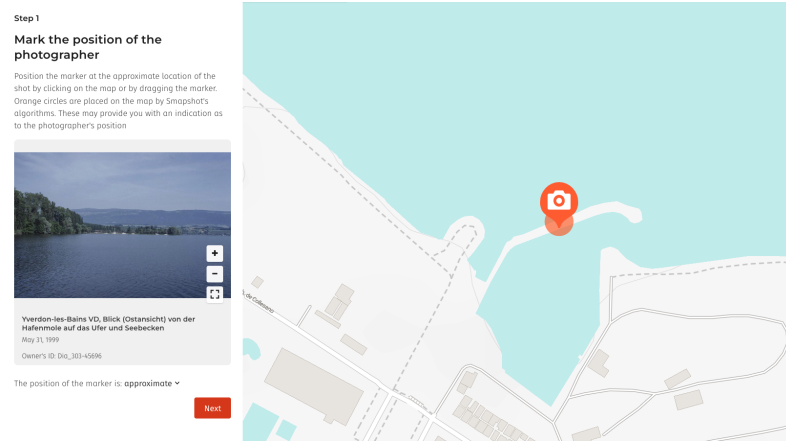


Figure 3.9: The user approximates the location of the camera [4]

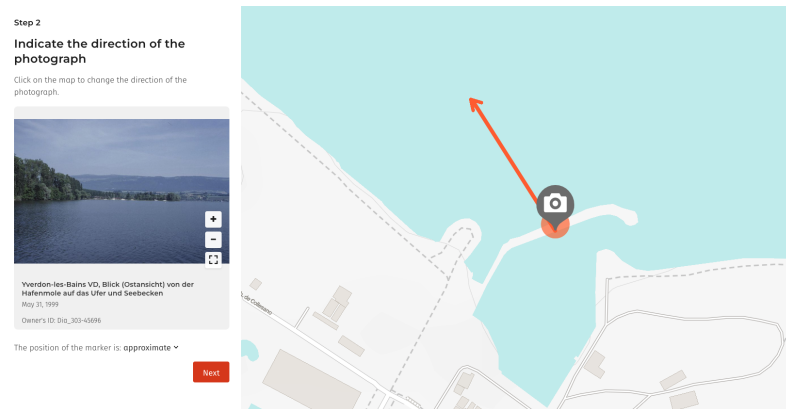


Figure 3.10: The user then specifies the direction of the camera [4]



Figure 3.11: The virtual globe is then shown and the user is asked to identify the GCPs, in the historic photo (to the left) and in the virtual globe (to the right) [4]



Figure 3.12: The position of the photograph is computed and placed aligned with the virtual globe

For the GCP in the 3D model the altitude can be extracted from the DEM. The GCPs are the inputs of a camera orientation algorithm, also called a 2D-3D pose estimation. The algorithm computes the unknown parameters of the camera [6].

3.4.1 Monoplotting

In Snapshot, the spatial georeferencing of images is done with a monoplotting algorithm. The inputs required are the digital version of the historical photograph, the DEM of the landscape, and the GCPs identified by the volunteers. In order to georeference, the monoplotting algorithm has to reproduce the optical system from where the photo was taken. What this means is to determine the location and orientation of the camera, the focal distance and the center of the images. When the position and orientation of the camera, and the focal length has been determined, the picture can be placed so that a line can be drawn from the projection center, passing through a selected point in the picture plane, intersecting the land surface of the DEM in the corresponding real point [36]. This is illustrated in Figure 3.13.

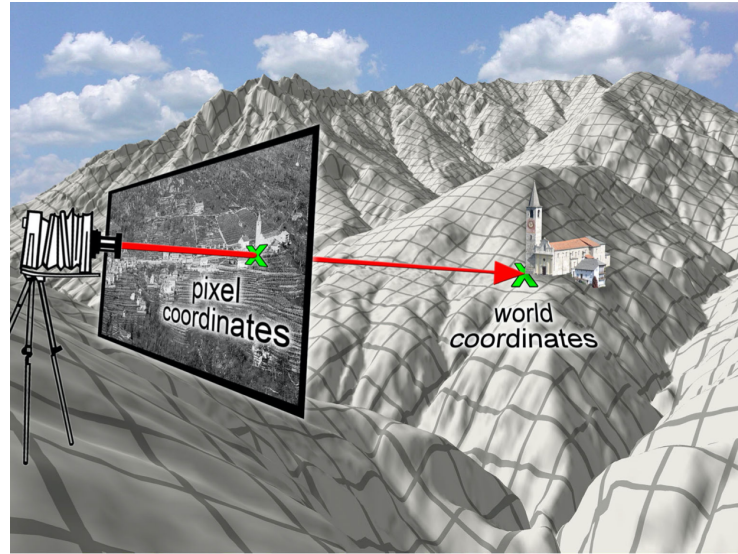


Figure 3.13: The principal of monoplottting [36]

The computation of the external orientation of the camera, meaning the location and orientation angles, is based on the collinearity equations commonly used in photogrammetry, see Figure 3.14 [36]. The equations relates coordinates in a sensor plane (from the 2D photograph), to object coordinates in three dimensions (the virtual globe) [37]:

$$x_a - x_0 = -f \cdot \frac{a_{11} \cdot (X_A - X_C) + a_{21}(Y_A - Y_C) + a_{31} \cdot (Z_A - Z_C)}{a_{13} \cdot (X_A - X_C) + a_{23}(Y_A - Y_C) + a_{33} \cdot (Z_A - Z_C)} \quad (3.1)$$

$$y_a - y_0 = -f \cdot \frac{a_{12} \cdot (X_A - X_C) + a_{22}(Y_A - Y_C) + a_{32} \cdot (Z_A - Z_C)}{a_{13} \cdot (X_A - X_C) + a_{23}(Y_A - Y_C) + a_{33} \cdot (Z_A - Z_C)} \quad (3.2)$$

where:

- x_0, y_0 are the image coordinates of the principal point.
- x_a, y_a are the image coordinates of an object A. These are the coordinates of the GCPs identified by the users of SmAPSHOT.
- X_A, Y_A and Z_A are the object's coordinates in object space.
- f is the focal length of the camera.
- X_C, Y_C, Z_C are the coordinates of the position of the camera in WGS84.
- a_{ij} are the coefficients of the rotation matrix, which defines the camera orientation. The orientation angles of the camera: azimuth, tilt and roll is embedded in the rotation

$$\text{matrix } R = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

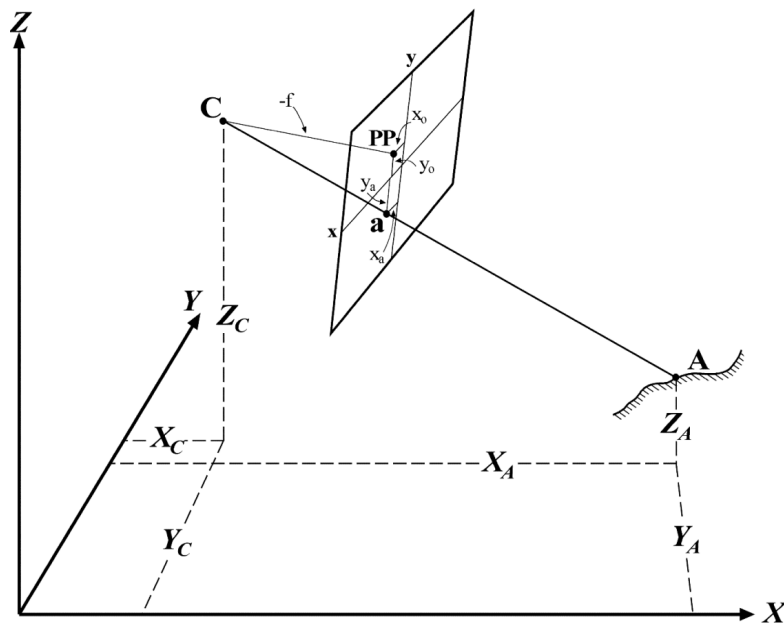


Figure 3.14: The Collinearity Condition [38]

3.4.2 Uncertainty estimation and validation in Smapshot

There are methods used to estimate the uncertainty of the georeferenced photographs in Smapshot. The first way is to simply calculate the number of GCPs. More identified GCPs results in a smaller error in the calculated parameters. Another way is to let the GCPs make up a bounding box in the picture. If the GCPs are scattered over the picture, meaning that the bounding box would cover a larger part of the picture, the error of the calculated parameters would be smaller. Furthermore, six GCPs are more than required in order to solve the equations, which means that an error can be calculated for each GCP by comparing the 2D coordinates digitized by the user, with the corresponding 2D coordinates computed with the algorithm [6].

Furthermore, after a photo has been georeference by a volunteer it is aligned with the virtual globe, and this is a good indicator of the accuracy of the location and orientation parameters. Additionally one step in validating the picture is based on the computed field of view. If the field of view is too large or too small the georeference will not be approved. The photo also has to be verified by advanced users of the Smapshot platform [6].

3.5 Rephotography

The difficulty of rephotography comes from the photographer having to disambiguate between the six degrees of freedom of 3D translation and rotation. One solution to this is to alter the viewpoint digitally, after the photograph has been captured [39].

One solution for rephotography is based on a technique where the user points a digital camera that is connected to a laptop, towards the scene depicted in some reference image, and the technique implemented estimates and visualizes the camera motion required to reach the desired viewpoint [7]. This visualisation of movement required is shown on the laptop. The technique implemented starts with the user loading the reference image, and then the user must photograph two photos. The first photo is to be captured with a slight angle away from the reference viewpoint. For the second photo, the user is asked to estimate the reference viewpoint by "eye". The system then computes a 3D structure from these two frames. The 3D structure is computed through the structure-from-motion technique, which is a photogrammetric range imaging technique. The user is then asked to identify six corresponding points (GCPs) between the reference image, and the 3D structure. Through the use of arrows the user is then instructed to the calculated reference viewpoint, and from there the user can recapture the historic photograph [7].

From two studies some challenges with rephotography were identified [7]:

- Communicating both 3D translation and rotation to user is challenging. The motion has six degrees of freedom, and the zoom adds a seventh degree.
- Sometimes the differences between a historical image can be very extensive due to, for example, architectural modifications, aging or time-of-day. These differences can make it difficult for the correspondences needed to compute the location and orientation of a photo.
- The cameras used to capture historical images are of sometimes of unknown calibration, for example a unknown focal length, or a noncentral principal point.

3.6 Data quality

The increase of availability of data enables it to be used in different types of applications, despite the fact the data quality can be unsuitable for such an application[40]. Both Smapshot and the mobile application created in this study rely on users with no formal background in spatial disciplines. Further, the mobile application is based on parameters computed through the georeferencing process of Smapshot, and it is not without error [6]. Naturally, neither are the real time parameters of the mobile device [20]. This raises the concern about the data on which the mobile application is based on, as well as the quality of the outcome of the rephoto.

3.6.1 Terms related to statistical data assessment

The total error is the difference between a measured value and the true value. The data containing the true values are referred to as reference data set. While the measurements, the set of the data from which the accuracy and precision can be calculated from, is referred to as test data set [40].

To summaries measurements of errors, accuracy and precision are used. Accuracy is how close the measured values are to the true value. Precision is how close the measured values are to each other. Standard deviations is a measurement of the variation of a set of values. Low standard deviation indicates that values tend to be close to the mean, and high standard deviation indicates that values tend to be spread out. The root mean square (RMS) is the square root of the mean square. [41].

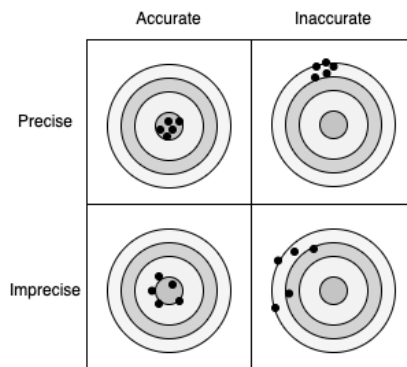


Figure 3.15: Accuracy and precision

3.7 Rephotography data quality in smartphones

3.7.1 Location and orientation

The manner of which a smartphone is held as well as device movement, has been found to impact the spatial accuracy, yielding greater errors than that of measurements performed on stationary devices. Additionally, being close to tall buildings further impact the accuracy. It is also worth mentioning that there is a tradeoff to be made high powered components and the accuracy, where expensive devices will results in more accurate measurements in regards to spatial location [27].

Most studies dedicated to assessing the spatial accuracy of GNSS sensors in mobile phones come to the conclusion that the accuracy is less than that of GNSS devices designed entirely with the purpose of navigation [42]. In a study by Clark et al., a smartphone of the model Iphone 4s was compared to a dedicated GNSS device of the model GPSmap60CSx. The relative error for the smartphone measurements were 7.86 m and for the GNSS device 1.69 m [43]. In another study the average positional error of a smartphone of the model Iphone 6,

was found to be 9.9 m [44].

In a study by Blum et al., the accuracy of the location sensing (GNSS) as well as the compass sensors, or geomagnetic field sensor as it is called in Android, were measured in situations designed to replicate real world situations. The measurements were carried out in varying environmental conditions in order to obtain data useful or understanding the practical limits of using these sensors when designing applications. The measurements that were taken in open areas resulted in a mean compass error of 10° . However, near buildings the it was found to be nearly 30° . The location error ranged from 10-30m, depending on the surrounding buildings. When walking close to skyscrapers a location error of 60 m was within the standard deviation [27].

3.7.2 Pitch and roll

In a study the accuracy of the pitch and the roll as illustrated in Figure 2.2, is investigated through the use of a device called the RollPitcher, see Figure 3.16. With this device, the phone can be mounted upon it, and the device allow for independent adjustment of tilt and roll through two different hinges. The objective status of both tilt, and roll could, with the help of the RollPitcher, be compared with measurements from native applications or via web browser [20].

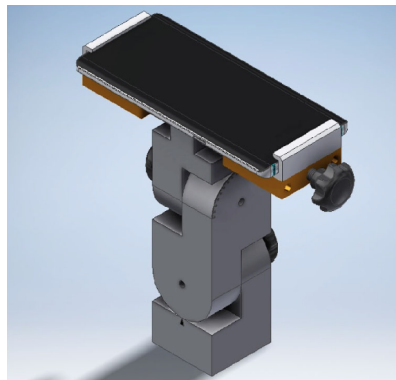


Figure 3.16: The Rollpitcher device [20]

The measurements of both tilt and roll were taken following a scheduled sequence of angle combinations. The tilt values were with help of the RollPitcher set to 0° , 30° , 60° , and 85° in both directions, and the tilt was then measured by a native application and a web browser. The same procedure was done but with the roll values. The scheduled sequence for the roll values were 0° , 15° , and 30° , also in both directions. These angles were chosen to simulate possible real life orientations of devices [20].

The result illustrated the inaccuracies for tilt and roll was usually between 2.1° and 6.6° . Deviations for roll were higher than for pitch. Furthermore, the inaccuracies differ depending on the smartphone device. The difference of measurements from native applications compared to those of a browser was small [20].

Chapter 4

Design Theory

4.1 Iterative design process

Design does not follow a linear path when it comes to the way of working. Instead, prototypes are being built, formed and re-formed several times over and over again [45]. This means that a design process is iterative and continuously improved as time goes on.

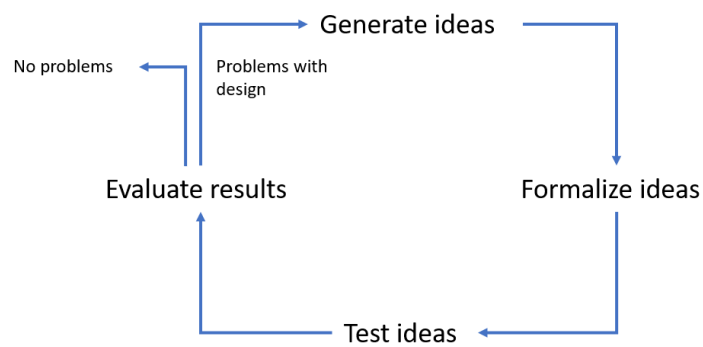


Figure 4.1: Illustration of iterative design process, inspired from Fullerton [46]

4.2 Brain- & bodystorming

To generate initial ideas about the mobile application, **brainstorming** was performed. Brainstorming is a technique used for generating ideas in order to find creative solutions for problems [47]. It can be used individually or in a group. There are two types of brainstorming sessions: structured and unstructured. Structured brainstorming has a clear agenda and ground rules in opposite of unstructured, where a group gets together without having any clear ground rules. An example of a ground rule can be that only one person is allowed to speak at a time or that no criticism is allowed.

The purpose of **bodystorming** is for designers to involve their bodies and simulate how the product could be used in a context [45]. Bodystorming is a technique often used in interaction design.

4.3 User-centered design

User-centered design (UCD) is a design approach where understanding users and user needs are the main focus [13]. In other words, UCD focus on each individual user and delivers design that is more user-friendly. A UCD process can vary depending on project and size. There are several techniques of involving users in the design by performing background interviews, questionnaires, work interviews, focus groups, on-site observations, role-playing and usability tests [12]. In addition, there is a template plan for UCD processes which goes hand-in-hand with Software Development Life Cycles (SDLC) [14]. The template plan contain following core components:

- **Team mission statement** - a short description that reminds the team of its purpose.
- **Project details** - answering questions of what is trying to be achieved.
- **User requirements** - research what potential users would require from the product.
- **Functional requirements** - what the application needs.
- **Diagrams** - useful if the application relies on datasets, e.g. dataflow diagrams.
- **Prototypes** - translating user and functional requirements into design that can be evaluated and tested.

It is not important to include all components in a UCD process listed above but to rather have a plan fleshed out before starting. This will mitigate the risk of spending unnecessary time on developing and designing features no one needs [14].

One way of identifying user requirements is to send out surveys. However the challenge with surveys is that it can miss extra information - whatever will not be asked will not be learned. It is therefore important to use careful survey design. One way to do this is by asking clear and concise questions as well as include a Likert scale. A Likert scale helps to collect feedback about how much a respondent agrees with a statement, by rating it from: *a. Strongly Agree* to *e. Strongly Disagree* [14]. It is possible to use social networks (family and friends) for reaching out and capture feedback from all ranges of experiences [14]. The next step after collecting information is to characterize the kinds of people that will be interacting with the design of a product [48].

4.3.1 Personas

A persona is a fictional character, used for representing the end-users when building an application [14]. Depending on what type of product, more than one persona can be created. The importance of a persona is to encapsulate the features of the people who will end up using the product [45]. By doing so, it can contribute to inspiration of the design. For example, a persona that is more playful and experimental might require a different design that satisfies his or her needs instead of a persona that is more orderly, careful and conservative.

The purpose of having personas is to develop and design a product that will work for the end-users. Personas are created in the beginning of a design process and is a part of the research phase. A persona contains specific information in order to focus on the features and details of the user group, such as: age, gender, experience level, goals and concerns. A few benefits of having personas is that focus will be increased on the needs of the users and it reduces fundamental changes to be made at the end of development. Moreover, it leads to better and more user-friendly design decisions [49].

Figure 4.2: Persona template from Xtensio, ©2022 Xtensio, Inc. [50]

Proto-personas

Proto-personas is a type of persona that is not based on any field research or observations of users [51]. Proto-personas can be used when there is scarce amount of time to research or when there is lack of financial support. Proto-personas are often based on secondary research or best guesses from a team and it is better to create a proto-persona than to have no persona at all. One benefit is that proto-personas enables rapid testing.

4.3.2 Prototyping

Prototyping is a way to visualize for users what is intended to be included in the product [14]. Prototypes can be shaped in a low quality (low-fidelity, also known as lo-fi) or high quality (high-fidelity, also known as hi-fi) manner. Lo-fi prototypes can be created by using sketch-based computer tools. These tools often allow quick prototyping and requires less skills than hi-fi prototyping would. On the other hand, lo-fi tools can limit the amount of interaction techniques in the prototype [52].

One advantage with lo-fi prototyping is that the test-user will go under the impression that little commitment has been made and therefore more encouraged to question the functionality and core concepts of the product [14]. Another benefit with lo-fi prototypes is that it is quick to sketch which gives more time to design between iterations [52]. The drawbacks of lo-fi prototypes are that it does not offer realistic interactions and clients might judge the prototype as unprofessional [52].

In complement to lo-fi prototypes, hi-fi prototypes have more realistic interaction and offers more design possibilities. However, hi-fi prototypes can make designers reluctant to fully explore the design possibilities [52].

4.4 Mobile interaction design

In mobile interaction design, there are three main components which can be applied in the purpose of creating a successful design for mobile applications [45]:

- **Understanding users** - develop an understanding of capabilities and limitation.
- **Develop prototype designs** - presenting a design that can be demonstrated and discussed.
- **Evaluation** - be able to refine the design by highlighting weaknesses and strengths.

Mobile interaction design includes three core components which also can be found in a UCD process. This means that mobile interaction design and UCD goes hand in hand. Some design starting points for mobile applications are to design for one-handed usage [45]. Another starting point is to take advantage of resources in the physical world.

4.4.1 Design principles

There are different design principles to consider when designing a mobile application.

Progressive disclosure

Progressive disclosure is a way to help users understand what is available and not in the product by hiding options that are not possible [14]. As an example of progressive disclosure, Microsoft Word grays out options that are not possible for the user to choose in a current situation and only highlights the options available, seen in figure 4.3.

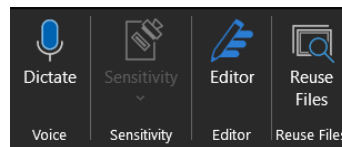


Figure 4.3: Progressive disclosure in Microsoft Words [53]

Principle of proximity

Furthermore, the principle of proximity, also known as the grouping principle or Gestalt principle, is a way to align text and information for it to make sense to the users [14]. Objects further apart seems to have less relation and by using this principle, the user's experience of using an application will improve significantly. The benefits of using principle of proximity are that organized layouts makes the user learn easier as well as it makes it simple to find things.



Figure 4.4: Example of the principle of proximity, where the right group is related

4.5 Gamification

Gamification is a way to make users motivated to use a certain product because it allows them to de-stress, have fun, socialize and achieve mastery [54]. Companies implement gamification in their products to keep the player motivated to use their product since it drives the outcome. There are several game mechanics to use. Points is a requirement for all gamified systems and there are several ways of implementing it: having cash score, video game score, social networking score or composite metrics [54]. It is also possible to include levels design and leaderboards. Smapshot's web-based platform is currently gamified, where users receive a rank based on their contributions (amount of georeferenced photos, observations and corrected titles or captions).

Participants

Last 30 days ▾ From 05 / 11 / 2021 ⓘ to 05 / 12 / 2021 ⓘ All collections ▾

Rank	Name ▾	Geolocalised images ▾	Observations ▾	Corrected titles or captions ▾
1	Walter Zweifel	466	3	15
2	Sigi Heggli	446	0	6
3	Urs Witmer	373	5	82
4	Hans Zumbühl	256	0	13
5	Daniel Vonwiller	181	1	9

Figure 4.5: Leaderboard of top five users on Smapshot as of December 2021 [4]

4.6 Design practices for maps

Michael Gaigg, a lead User Interface Engineer, presents design patterns for developers to use when designing a map application used for navigation [55]. As an example, the "Blue Dot" solves the problem of displaying for the user their current location on a map to help them navigate their surroundings. A suggestion by M. Gaigg is to only display useful information to the user on a map by avoiding useless data like longitude or latitude coordinates. Another design pattern is to have zoom control where the user can zoom in or out in order to see a more detailed view of the map.

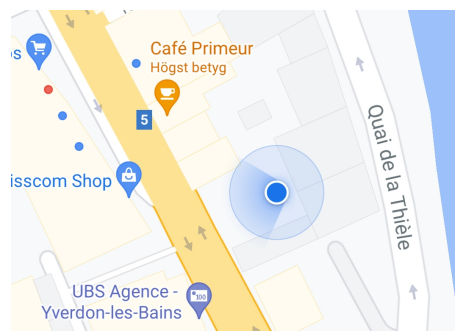


Figure 4.6: Blue dot from Google Maps, map data ©2022 Switzerland [56]

4.7 Usability testing

Usability testing refers to a process where participants are recruited, representing the end-users in order to evaluate and determine if a product meets the intended usability criteria [57]. Usability testing can be seen as a research tool and the amount of tests to be performed can vary, depending on the sample size of test-users. The goal of carrying out usability tests is to expose design problems and therefore eliminate or minimize frustration for users.

4.7.1 Think aloud

Think aloud is a technique used in usability testing in order to capture the test-users thought process when making decisions in using a product [57]. A benefit of think aloud is that it shows how a test-user is trying to work around a problem. It also gives clues to how a test-user is thinking about a product and whether the design matches the way the product works. On the other hand, test-users always filter out certain things as they speak which means that it is impossible to capture everything that is going on in a test-users mind. Think aloud also slows down the time to complete a task, which affects measuring of time.

4.7.2 Remote usability testing

It is possible to perform remote usability testing with computer-to-computer tools such as video conferencing. With speakers, an administrator can capture comments and think aloud technique from the test-user and record facial expressions [58]. One study compared traditional usability testing with remote usability testing and the results showed that more usability problems were discovered by remote usability testing even though the usability problems discovered in both cases were similar [59]. The reason for this is that by having test-users performing tasks at home they can also capture issues unique to their environment [60].

4.7.3 Number of participants

The amount of users depends on what is being planned to achieve, but usability expert Jakob Nielsen suggests a minimum of five users. The reason behind the number five is because the first five test-users will discover majority of usability errors, up to 85% of usability errors [14]. Having more than five test-users can make the study complex and troublesome. Moreover, the range between five to 10 test subjects will produce a meaningful insight to what can be improved.

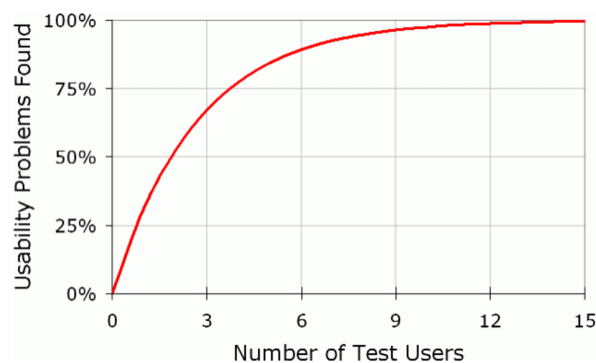


Figure 4.7: Number of test-users in relation to usability problems found [61]

4.7.4 System Usability Scale

A System Usability Scale (SUS) is one example of a questionnaire to gather data about how satisfied participants were with the product they used in a usability testing session. It was developed in order to evaluate all topics regarding usability testing according to ISO standard 9241 [57]. SUS emerged from the need of having evaluation methods that were simple and quick, instead of the existing exercises that could last between 20 minutes to one hour [62]. SUS is measured as a score and more than 68 is considered above average [63]. The max score of a SUS is 100.

The image shows a screenshot of the System Usability Scale (SUS) survey. It consists of three separate items, each with a 5-point Likert scale. The scales are labeled 'Strongly disagree' on the left and 'Strongly agree' on the right, with numerical markers 1, 2, 3, 4, and 5 in between. Each marker is accompanied by a radio button.

1. I think that I would like to use this system frequently.

1 2 3 4 5

Strongly disagree Strongly agree

2. I found the system unnecessarily complex.

1 2 3 4 5

Strongly disagree Strongly agree

3. I thought the system was easy to use.

1 2 3 4 5

Strongly disagree Strongly agree

Figure 4.8: SUS survey, used in the design process

Chapter 5

Design Process

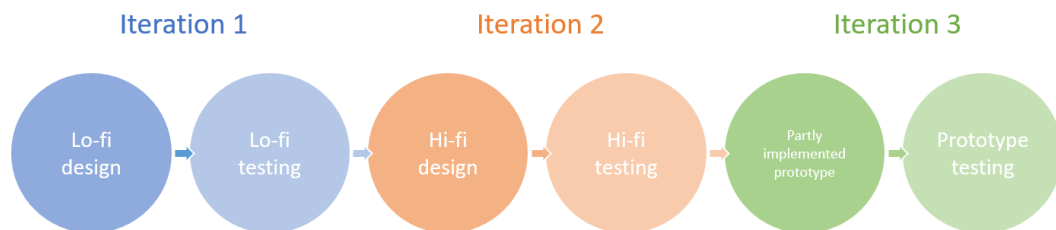


Figure 5.1: Overview of design iterations carried out

5.1 First iteration

The first iteration highlights the start of the design process. It lasted for five weeks and focused on literature research and generating ideas about the mobile application, its design and features. The first iteration also consisted of collecting data about end-users and designing a lo-fi prototype to evaluate.

5.1.1 Brain- & bodystorming

The initial meetings sparked an unstructured brainstorming session in order to create a common picture of what was to be defined, planned and carried out. This meeting involved the supervisors from Switzerland as well as a developer from the Snapshot project. After the brainstorming session, bodystorming was carried out. It was performed outside with an Android phone (Xiaomi Mi 9T) in Yverdon-les-Bains, Switzerland. The bodystorming session contributed to structuring the workflow of the mobile design:

1. Select a photograph from a list.
2. Navigate to the location of the photograph.
3. Retake the photo.
4. Compare the two photos.

Bodystorming was performed in Yverdon-les-Bains by one person and observed by another. The person simulated the behaviour of selecting a photo to recapture in an outdoors environment. It raised ideas and thoughts regarding the first step of the workflow, such as having filtering options for the user to select specific types of photos to retake and setting distance to how far the photos can be from the user's current location. Another idea was also raised: having a pre-planned route for the user instead of having to find and select photos to retake manually.



Figure 5.2: Bodystorming performed in Yverdon-les-Bains

5.1.2 Storyboarding

Storyboarding is used to visualize an early mock-up design of the mobile application and is more visual than having dataflow or workflow diagrams [14]. During the research phase, storyboarding was used to sketch a simple view of the design scope of the mobile application.

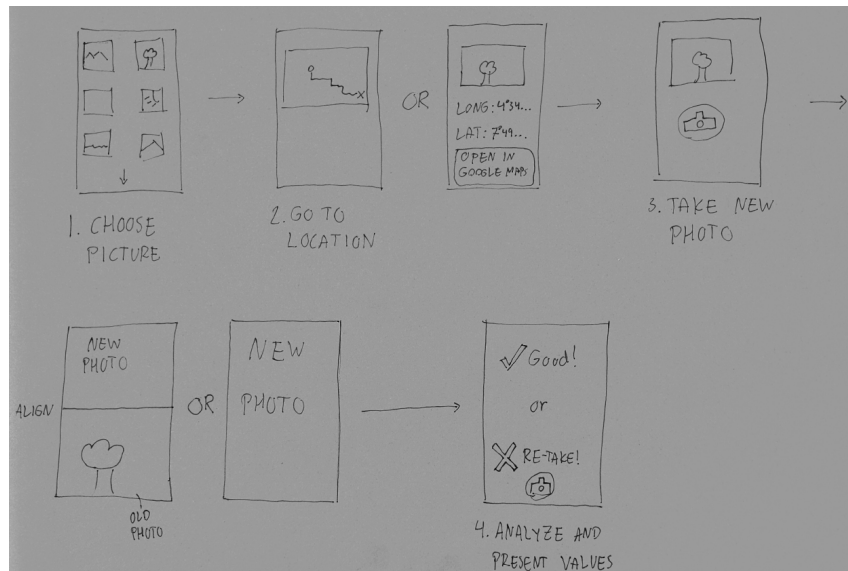


Figure 5.3: Storyboarding the design of the mobile application, skissed on a paper

5.1.3 Survey

The reason for conducting a survey study was to collect information about who the potential end-users might be and to see if there is any interest of using a Smapshot mobile application. The aim was to collect information from volunteers from Smapshot's web-based project as well as reaching out to social networks. In order to receive feedback from as many potential end-users as possible, the survey was translated into: English, French and German. Surveys provide a quick way of gaining feedback and is rather cheap in comparison to other data-collecting techniques such as interviews [45]. The survey was sent out to 100 Smapshot volunteers subscribed from their newsletters and an additional 40 people from social networks: friends, family and colleagues. 52 out of 140 people answered the survey in total with 19 females and 33 males participating. The results were as following:

- Smapshot volunteers: 38 answered out of 100 (38%).
- Social networks: 14 answered out of 40 (38%).

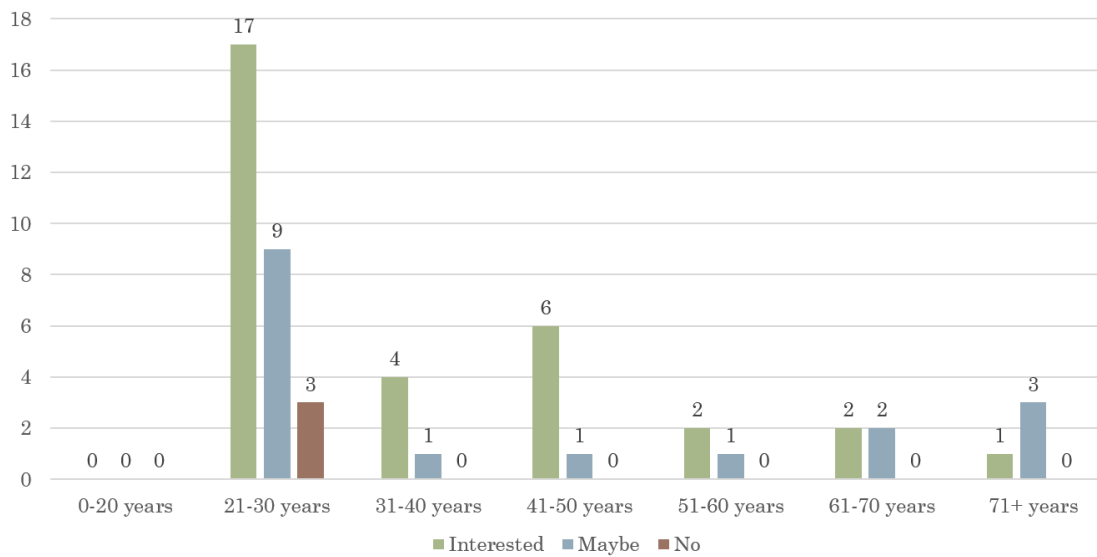


Figure 5.4: Age distribution and interest data collected from the survey

5.1.4 Defining end-users

The survey results showed that there is an interest among people from all ages. Out of the 52 people who answered, 37 of those who answered lived in a city and 31 were working. On average, people rated themselves high when it comes to experience in using mobile applications: on a scale from one to 10, the average rating was 8.6. The comments received from the survey was:

"Would maybe be fun to use the app one or two times."

"How do I stay motivated to use this app? Could use campaigns or rankings perhaps."

"Cool to use abroad but not at home. Would love to use when on vacation or sightseeing."

"I would need a big screen for this, not a small mobile screen."

"Not sure when I would use it."

5.1.5 Personas

After data collection from the survey, the first persona was created: Marc Schwyn. He is based on the biggest potential user group (21-30 years old) from the survey statistics that had an interest in using a Smapshot mobile application. From this user group, 17 were interested, nine answered "maybe" and three answered "no". It is also the only user group that had three people answering "no" to the question of interest in using a Smapshot mobile application.

Marc Schwyn



Bio
Marc has recently graduated from university and has been working for about 2 years. He lives with his partner and they enjoy exploring new places together. A part from that, Marc spends much time on his phone and is rather tech-savvy.

Frustrations

- Lack of motivation
- Losing interest quickly
- Often busy

Goals

- Explore
- Travel

"I like to try new things."

Age: 28
Status: working
Location: city, apartment

Figure 5.5: First persona, based on survey data

The second persona, David Berg, is based on the second largest user-group that are interested, which is the age group of 41-50 where six people were interested and one answered "maybe". David Berg was improved after receiving further user feedback from the lo-fi tests: one of the lo-fi test-users were not interested in gamification but rather the history behind the photographs to retake.

David Berg



Bio
A part from work, David spends his time on having several side projects. David lives a very active lifestyle. He enjoys exploring, learning and experiencing new things in life.

Frustrations

- No interest in gamification
- Lack of time, prioritize hobbies

Goals

- Having side projects

"Life is all about experience."

Age: 45
Status: working
Location: village, house

Figure 5.6: Second persona, based on survey data and user feedback from lo-fi testing

A third persona was created: Louise Hoffman. This persona's age and gender is based on the survey statistics. The rest of the persona traits are guesses, therefore making it a proto-persona.

Louise Hoffman



Bio
Louise is a student who travels a few times each month to go hiking. She is ambitious and practice climbing on her spare time as well as skiing during the winter.

Frustrations

- Complicated instructions
- Don't like being spontaneous

Goals

- Go hiking few times a month
- Exercise sports

"I enjoy being outdoors!"

Age: 23
Status: student
Location: city, apartment

Figure 5.7: Third proto-persona

5.1.6 Scenarios

Scenarios are used for enhancing the experience of a user when using the application, evaluating the design decisions and determine if they complete the user's needs [14]. After the first persona was created, scenarios were set up in order to think about the different situations the personas might be in. Following scenarios were created:

S1. Louise is hiking and wants to take pictures of the nature. How can she accomplish this by using the application? This scenario is covered by having the feature of retaking photos and having a camera view in the mobile application to capture photos.

S2. Marc has used the application a few times but finds no motivation in continue using it. How can the application motivate him? By receiving points as a gamification feature, motivation can be encouraged for further usage.

S3. David has a bad habit of not being accurate when recapturing photos. How can the app motivate and assist him in being more precise? In order to mitigate the risk of having bad habits, instructions on how to retake a better photo can be provided to the user.

5.1.7 Low fidelity prototype

A lo-fi prototype was created as a mock-up, using Microsoft PowerPoint [64]. The lo-fi prototype is based on brainstorming, bodystorming, storyboarding and the survey data.

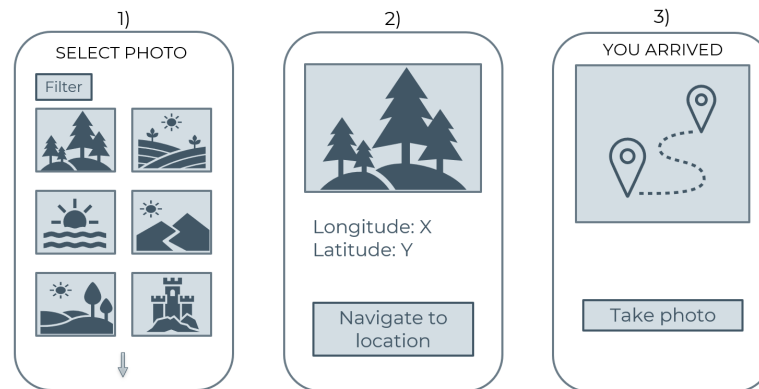


Figure 5.8: Lo-fi prototype with initial steps

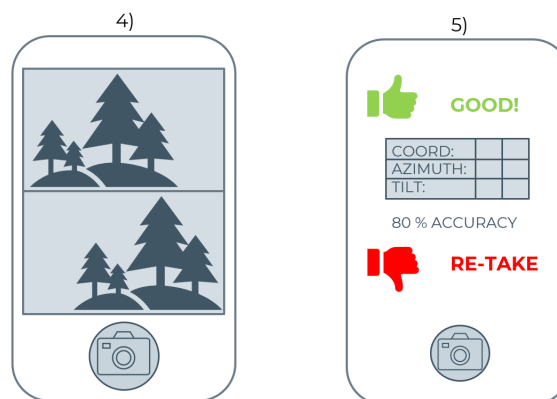


Figure 5.9: Lo-fi prototype with the last steps

Low fidelity testing

Lo-fi testing were setup in a remote testing environment similar to remote usability testing but not as formal. Two test sessions were performed with two test-users, both males in their mid 20s. The test sessions consisted of live sharing the mock-up design as a presentation. The test-users connected through Zoom [65] and pretended to click on their computer screen in an attempt of simulating an interaction with the presentation slides. Web cameras and microphones were used to capture the test-users expressions. The reason for only having two test-users was because to not spend too much time on the first iteration. After the testing, an informal interview was conducted with each test-user to capture preference data and potential design flaws in the lo-fi prototype that can be improved.

The test-users expressed following thoughts and ideas about the lo-fi design:

- **Filtering options** - would be nice to sort on different types of photos, such as landscapes or cities. Also be able to set a walking distance range when selecting photos nearby.
- **Transparent camera view** - it is hard to remember the selected photo when taking a photo and would therefore prefer to have the photo transparent on top of the camera view.
- **Better result view** - give instructions on how to achieve better accuracy instead of displaying parameters (azimuth, coordinates and tilt). Display a general score instead.
- **Information about the photos** - include information about the old photos such as when the photo was taken, from where and by who.
- **New feature** - include a feature to give suggestion of photos to retake nearby and the time to walk the designated route.

5.2 Second iteration

The second iteration consisted of creating a hi-fi design using the software tool Figma [66], technical development of the mobile application as well as performing the first run of usability tests of the hi-fi design. It lasted for seven weeks in total.

5.2.1 High fidelity prototype

The hi-fi prototype is based on the results from the previous lo-fi testing and workings of the first iteration. Figma allows interactions with the design on any type of device which also means that the hi-fi design can be tested on mobile devices. The color scheming and the text font montserrat is based on Smapshot's website [4].

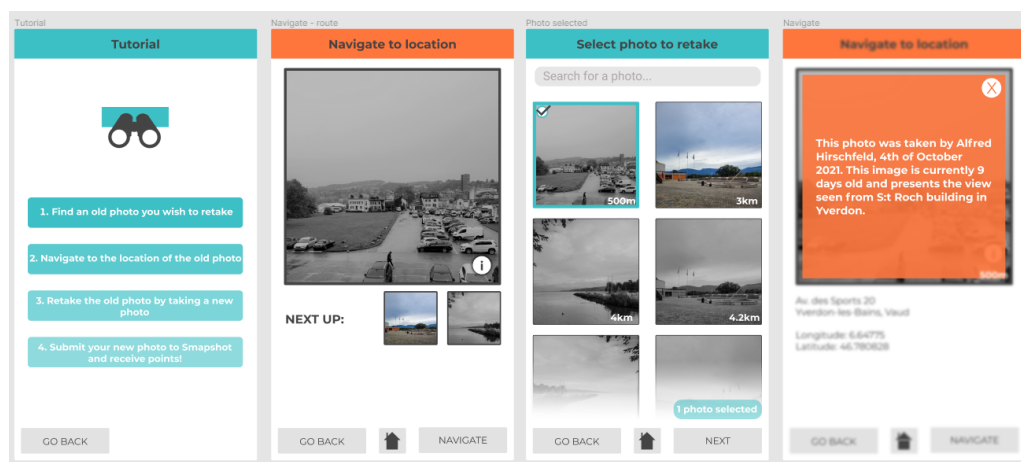


Figure 5.10: Screenshots from the hi-fi prototype

"Challenge of the day" was a new feature that was set up as a pre-planned route including the three closest photos to the user, instead of having to find and select several photos manually. This feature was based on the feedback from the lo-fi testing in the previous iteration.

The filtering options include setting a distance as a range of how far away a user wants to see old photos from his or her location. In addition, the user can also include certain type of photos to display, by filtering photos from cities or landscapes, as expressed from the user feedback. All photos used in the hi-fi prototype are mock photos and not photos from Smapshots website.

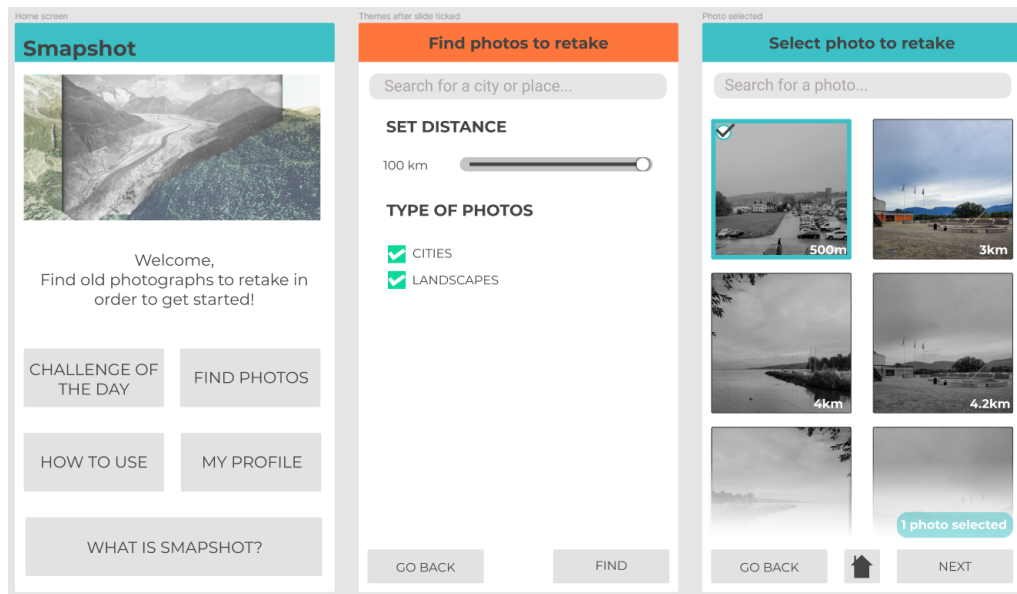


Figure 5.11: Screenshots of the hi-fi prototype showing the start screen, filtering options and photos to retake

When opening the camera to retake a photo, the user can select between crop mode and transparency mode. Crop mode, seen in figure 5.20, is a way for the user to crop and minimize the camera view in order to help the user to retake a photo. Crop mode gives support for aligning the camera view with the edges of the selected photo which is useful when capturing landscape photos. Transparency mode allows the user to place the selected photo on top of the camera view in order to facilitate retaking a photo. The phone's real-time orientation angles (azimuth, tilt and roll) are displayed in the upper left corner of the camera view. These orientation angles are compared with the corresponding orientation angles of the selected photo in order to calculate a result of how accurate the recaptured photo is. The hi-fi prototype also allows the user to submit the newly taken photo in order to be placed on a leaderboard against other made-up users, similar to the leaderboard being shown on Smapshots website, figure 4.5.

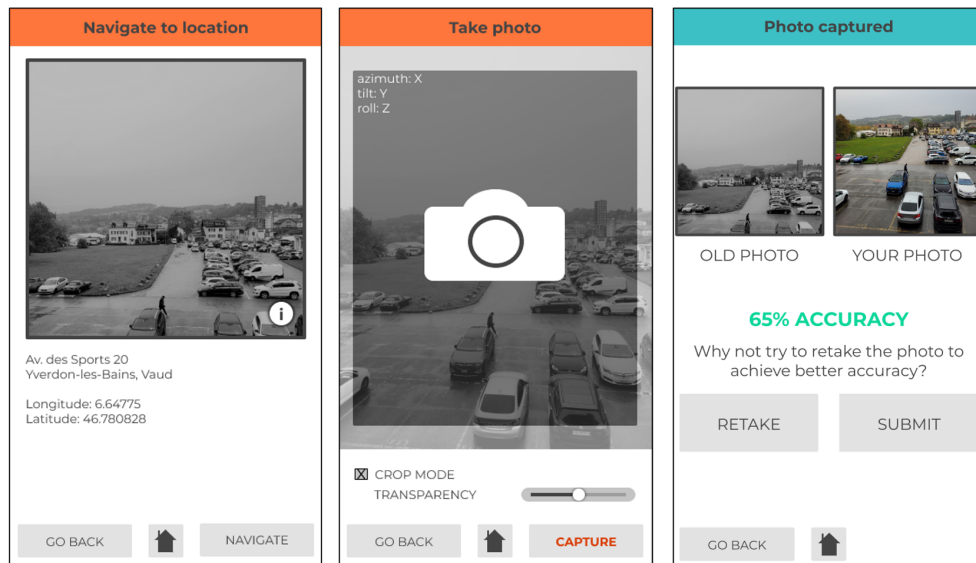


Figure 5.12: Hi-fi design showing the hypothetical camera view with transparency and cropping mode toggled as well as the results

5.2.2 First usability testing

Setup

A usability test plan was created in order to specify the purpose of the usability testing [57]. The purpose was to identify pain points in the current design, find out if the current features are useful and if anything is missing. The test plan included: research questions, non-disclosure agreement for testing, background questionnaire, orientation script, tasks for the test-user to perform, a SUS survey and a post-interview. There were three tasks set up for the test-user to complete:

A1) Find and select a photo to retake.

B1) Improve the accuracy of your newly taken photo by retaking it.

C1) Use the pre-designed route for capturing photos nearby ("Challenge of the day").

Participants

Five test-users were recruited: four male and one female. The ages ranged from 23 to 33 and all test-users rated themselves high when it comes to experience in using mobile applications. The test-users covered the first persona Marc Schwyn (figure 5.5), as well as the proto-persona Louise Hoffman (figure 5.7). In addition, the test-users covered the biggest potential user group (20-30 years old) as well as the third biggest user group (30-40 years old) according to the survey statistics.

Table 5.1: Demographics of test-users from first usability testing

Test-user	Gender	Age	Experience (1-5)
T1	Male	25	5
T2	Male	30	5
T3	Female	25	5
T4	Male	23	5
T5	Male	33	5

Procedure

Before conducting the tests, the test-users had to sign a non-disclosure agreement and answer a questionnaire that captured their gender, age and experience in using mobile applications on a scale from one being the lowest to five being the highest (very experienced). Throughout the tests, the test-users were observed. Time to complete a task from each test-user was measured and captured as performance data to raise an understanding if the tasks were too complicated for the user. The max amount of time to complete a task was based on a pilot test. The tests were performed indoors at the Stähli residence in Yverdon-les-Bains as well as at the St Roch office of HEIG-VD. The host acted as both test moderator and observer at the same time. The advantage of being indoors is that it was easier for the host to observe. Also, the test phone used could be connected to wifi indoors since it did not have a SIM card.

Results

After performing five usability tests, the results were as following:

- **Missed the orientation angles** - all five test-users failed to notice the azimuth, tilt and roll in the upper left corner of the camera view.
- **Points are motivating** - all five test-users answered that the points received after taking a photo motivated them to perform better.
- **Transparency mode** - all five test-users thought that the feature of having transparency of the selected photo on top of the camera view made sense.
- **Crop mode** - three test-users did not understand the crop mode feature.
- **Tutorial instructions unclear** - two test-users found the "How to use" instructions unclear due to misinterpretations.
- **Display a map** - two test-users would prefer to have a map for selecting photos in addition to selecting from a list.
- **Zoom function** - two test-users wished for a zoom function with the photo in the camera view.
- **Filtering options** - one test-user wanted more options for finding photos such as filtering by monuments and rivers. Another test-user pointed out that setting distance did not make sense because you are only interested in photos close to your location.
- **"Challenge of the day"** - one test-user thought that this feature was too similar to the main functionality of the mobile application.
- **Bigger button** - one test-user suggested that the button for finding photos should appear as the biggest button on the main menu since it is the main feature of the application.
- **Social involvement** - one test-user suggested to involve the users more socially, such as sharing photos with friends.

Table 5.2: Time performance from each test-user

	Task 1 (4 min max)	Task 2 (2 min max)	Task 3 (2 min max)
T1	2:15	0:40	0:43
T2	3:07	1:20	1:15
T3	2:45	1:00	0:45
T4	3:45	0:57	1:20
T5	1:15	0:45	1:30
Avg.	2:37	0:56	1:07

The SUS score received was: 77.5, 72.5, 90, 70 and 80. The average SUS score was 78.

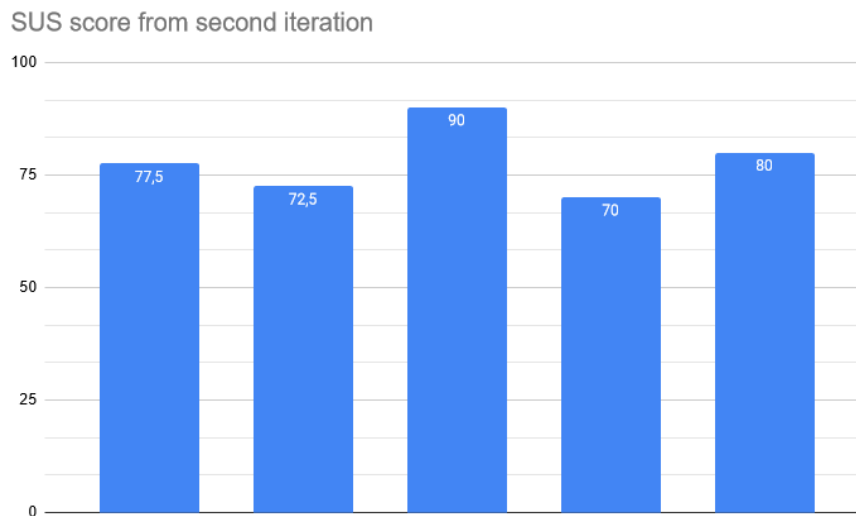


Figure 5.13: SUS score from second iteration

5.2.3 Mobile development

In addition to the usability testing, the second iteration also consisted of developing the mobile application. The ambition of the project was to usability test a final prototype in the third iteration. A custom camera view was implemented to be able to develop the transparency feature. One test photo from the office building at St Roch in Yverdon-les-Bains was added to the mobile application in the purpose of testing. The test photo had to be uploaded to Smapshot's beta website and georeferenced in order to include it as a test photo. A result view was implemented to display a score where the orientation angles of the photos are being compared.

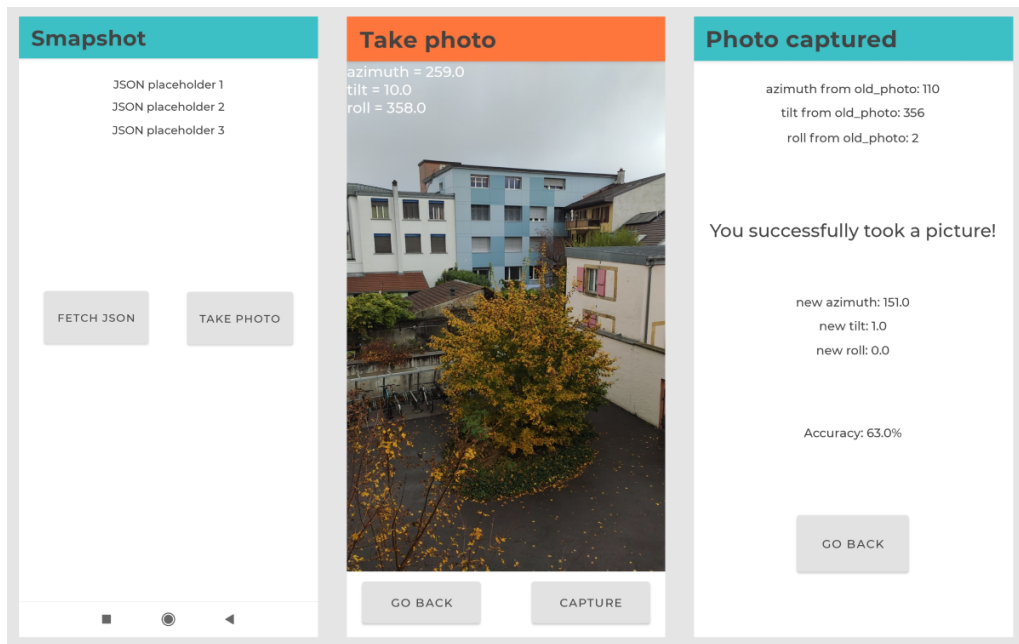


Figure 5.14: Custom camera and a result view of taking a photo

5.3 Third iteration

The majority of the third iteration consisted of mobile application development and usability testing in order to evaluate the final design decisions made. It lasted for seven weeks in total.

5.3.1 Partly implemented prototype

For the third iteration, the main focus was to develop a mobile application based on user feedback from the second iteration. Due to the lack of time, a partly implemented prototype was developed. Both progressive disclosure and principle of proximity was used in the design.

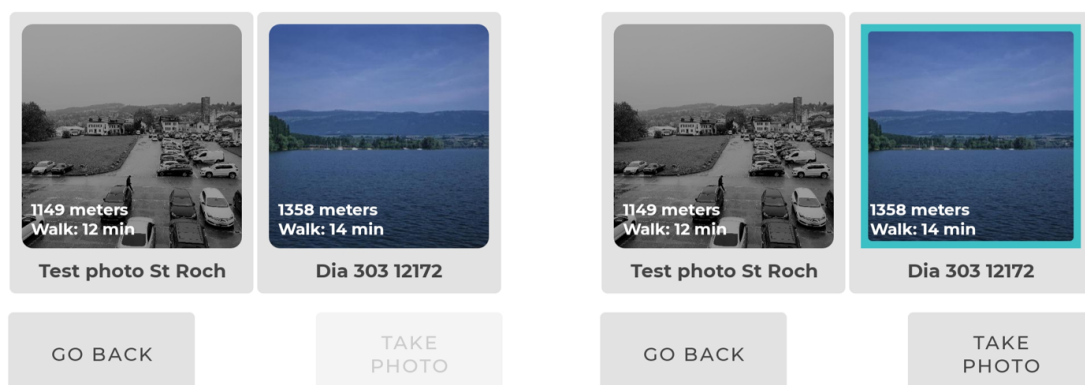


Figure 5.15: Progressive disclosure, where the user has to select one photo in order to take photo

A zoom function for the transparent photo was implemented so the user could zoom in and zoom out the selected photo. "Challenge of the day", the feature which plans a quick route for the user of photos nearby, was removed for two reasons. The first reason was that one test-user thought it was too similar to the ordinary functionality of the mobile application. The second reason was that more time would be needed to spend on developing and therefore deprioritized. Filtering options were improved and more options were added: cities, landscapes, monuments, rivers and viewpoints. The filtering options are not implemented technically because it is not possible to categorize photos from Snapshot after landscape photos, cities, rivers, etc. Setting a range of distance was removed.

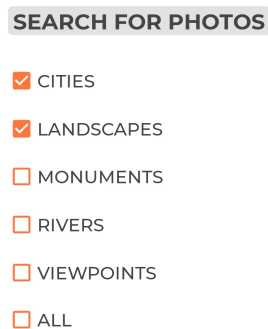


Figure 5.16: Filtering options when searching for photos to retake

When selecting a photo, a map view from OpenStreetMap is displayed with a blue marker that shows the current location as well as the location of the photos to retake nearby. A distance and time estimation to walk is also displayed. The distance is calculated as an absolute distance between current location and location of the photos. Absolute distance is defined as the shortest distance in meters between two separate locations [67]. The time estimation is based on an approximation that one kilometer equals to 10 minutes of walking. When selecting a photo, information about that photo is displayed, seen in figure 5.17. There are two photos to retake in Yverdon-les-Bains, the test photo and one photo from Snapshot.

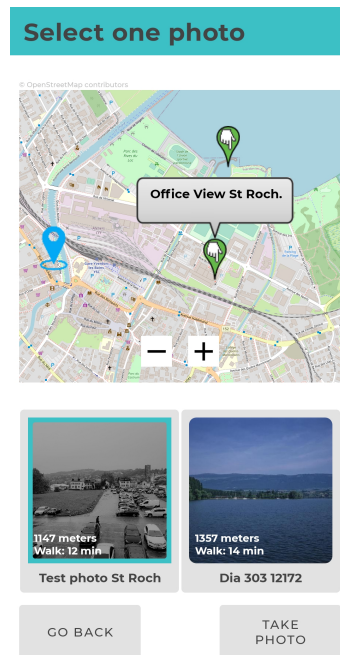


Figure 5.17: Selecting a photo to retake

The instructions on how to use the mobile application were improved in an attempt to be written more clearly for the user to interpret.

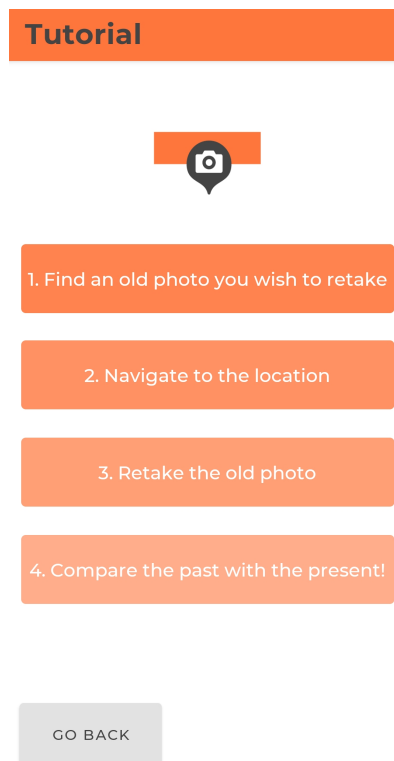


Figure 5.18: Instructions on how to use

To prevent the users from taking the photo at the wrong place, a distance message is displayed on the camera view. This message only displays if the user is more than 20 meters away from the location.



Figure 5.19: Message displayed on the camera view if the user is too far away from the location

After a meeting with the supervisors from Lund, the decision was made to have grading instead of points that was based on the accuracy of retaking a photo. The reason for this is because it is not possible to base points on the accuracy, since the accuracy has too many sources of errors. One source of error is the uncertainty of smartphone sensors and the other error is the uncertainty from Smapshot's georeferenced photos. The grading score, ranging from A to F is based on the difference of the orientation angles.

5.3.2 Second usability testing

Setup

The second usability tests were carried out indoors, in Sweden with one test photo taken in Lund and another in Malmö, replacing the test photo previously used from the office view in St Roch, Yverdon-les-Bains. A second usability test plan was created, based of the first test plan and contained three tasks for the user to perform:

- A2) Understand what Smapshot is and how to use the app.*
- B2) Retake the closest photo.*
- C2) Try to improve your grade by retaking the same photo again.*

Participants

Due to covid-19 outbreak in December 2021 and January 2022 with omicron, three test-persons were recruited. According to figure 4.7 three test-users will still expose more than the majority of usability errors.

Table 5.3: Demographics of test-users from third design iteration

Test-user	Gender	Age	Experience (1-5)
T1	M	62	5
T2	M	23	5
T3	M	25	5

Procedure

The test-users signed a non-disclosure which was the same one used in the previous iteration. The usability tests were performed in-person with one test-user and the host acting as test moderator and observer for each test-session.

Observations

A2) Understand what Smapshot is and how to use the app.

None of the test-users did task A2) correctly and help was given in all cases. All test-users started with pressing on the button "Find photos". One test-user pointed out that the introduction text "Welcome - find old photos to get started" told him to press on "Find photos" as first step. When reading the instructions of how to use the app, one test-user thought that "navigate to location" meant to open Smapshot's website and navigate to the current location. Another test-user thought that he would place a recaptured photo on a map. Two test-users believed that it was possible to press on the different red boxes with text on, seen in figure 5.18. One of those test-users pointed out that tutorial is the wrong word to use under instructions since it misleads the user.

B2) Retake the closest photo.

During task B2), the first test-user believed that he was supposed to select a photo from Smapshot's website. Later on, when opening the camera view, he focused on the green parameters in the upper left corner and did not use the transparency feature. Both first and second test-user did not use the transparency slider when capturing a photo.

C2) Try to improve your grade by retaking the same photo again.

Two test-users used the transparency slider on the second attempt of retaking a photo.

Results

- **Simple and easy to use** - all three test-users found the user interface simple and easy to use.
- **Map makes sense** - all three test-users thought that having a map to select photos from and displaying your current location makes sense.
- **How to use instructions unclear** - all three test-users found the "How to use" instructions unclear, due to misinterpretations of the instructions. One test-user suggested that it is probably easier to learn by doing instead of reading instructions.
- **Confusing orientation angles** - two of three test-users did not understand the angles azimuth, tilt and roll and one test-user expressed them to be repulsive. However, they did understand that they were supposed to be green. One test-user suggested to replace the parameters with a floating dot on the center of the camera view and provide visual feedback for how the phone should be tilted and held.
- **Less motivating effect of grading** - two test-users strongly disagreed that the grade received from retaking a photo motivated them to perform better. One test-user was neutral to the statement.
- **Usage of application** - two test-user answered that they would not use the mobile application regularly (more than once every week), and instead they would use it occasionally (every two weeks or less). The remaining test-user would neither use the application regularly nor occasionally.
- **Profile feature** - one test-user wished for saving captured photos to a profile.
- **Pinch to zoom** - one test-user wanted the "pinch to zoom" instructions to appear on the camera view as an animation to make it more obvious.
- **Make transparency more obvious** - one test-user did not use the transparency feature at all and did not perceive it as a slider. All his attention was focused at making the orientation angles green. Therefore, he suggested to add an animation for the transparency slider to make it more obvious.
- **Involve social media** - one test-user suggested to involve social media such as sharing on Facebook your recaptured historical photos. He compared it to the application Runkeeper [68], where people can share their runs and other statistics with friends.

Instead of measuring time as previously done in the first usability testing session, amount of help was captured as performance data.

Table 5.4: Amount of help needed by each test-user

Test-user	Task A2	Task B2	Task C2
T1	1	1	0
T2	1	0	0
T3	1	0	0

The SUS score received was: 67.5, 80 and 75. The average SUS score was 74.

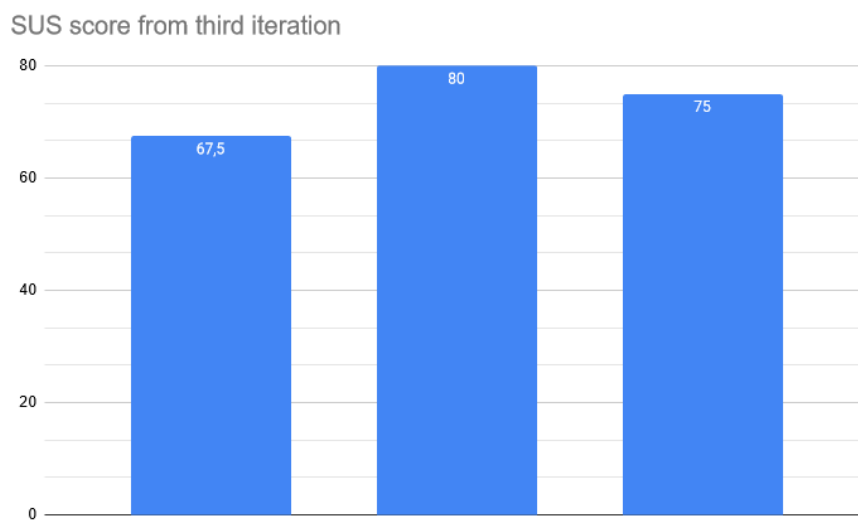


Figure 5.20: SUS score from third iteration

Chapter 6

Development implementation

The mobile application is developed for Android, using Android Studio and agile software development. Android's custom camera API was used to implement two techniques for enabling support of rephotography.

6.1 Technical Background

Android

Android is a mobile operating system. Android mainly focuses on touchscreen devices. The official integrated development environment (IDE) for Android is called Android Studio. Kotlin is Google's preferred language for application development with Android Studio, however both Java and C++ is supported [69].

JSON

JSON (Javascript Object Notation) is a text based format that is language independent, and used to exchange data. A JSON file contains JSON-objects that are made up of a String within quotations and a corresponding value that can be a String, Integer, another JSON-object, Array, Boolean or Null [70].

When georeferencing a photo in Smapshot, a JSON file is generated which stores all information about the photo. It stores information such as: author, title, year the photo was taken, coordinates in longitude and latitude, azimuth, tilt and roll parameters and more.

```
apriori_altitude: null
id: 185747
is_published: true
original_id: "Dia_303-12172"
▼ title: "Yverdon-les-Bains VD, Blick (Ostansicht) von der Hafemole auf das Ufer und Seebecken"
▼ license: "ETH-Bibliothek Zürich, Bildarchiv / Fotograf: Lüem, Peter / Dia_303-12172 / CC BY-SA 4.0"
download_link: "http://doi.org/10.3932/ethz-a-000414748"
link: "http://doi.org/10.3932/ethz-a-000414748"
shop_link: null
observation_enabled: true
correction_enabled: false
state: "validated"
view_type: "terrestrial"
width: 1500
height: 1185
date_shot_min: "1999-05-31"
```

Figure 6.1: An example of a JSON file from a Smapshot photo, containing keys (blue) and values

6.1.1 Selection of photos

A test photo was captured from the office building of St Roch, HEIG-VD in Yverdon-les-Bains to be included in the mobile application. This photo was uploaded to Smapshot Beta, georeferenced and downloaded alongside its generated JSON file containing all information about the photo. A black and white filter was added to the photo in order to mimic the feeling of an old photo. Another terrestrial photo, captured in 1999 in Yverdon-les-Bains from Smapshot's official website, was georeferenced and downloaded to use as the second test photo in the mobile application.



Figure 6.2: The two photos used in the mobile application, left being the St Roch photo and right being the photo from Smapshot

6.1.2 Application programming interface

Application programming interface, commonly known as API, makes data and services public through a set of assets, such as objects and methods [71]. The advantage of using APIs to fetch data from is that it is possible to do so without having to implement the basic procedures. APIs are essential to modern software since they facilitate programming tasks and supports reusable code. When using APIs, it is possible to include a URL of a service to reach its endpoint. An endpoint is the end of a communication channel that points to a location of a resource on a server such as a URL, and APIs use endpoints to interact with over systems.

Custom camera API

The Android framework allow developers to use custom camera features when developing applications for creating custom camera experiences [72]. Android's custom camera API is used in this mobile application in order to be able to use transparency mode on top of the camera view as well as displaying real-time parameters of azimuth, tilt and roll when capturing a photo.

Smapshot API

Smapshot's API is used to fetch data and images of old photos to visually display them in the mobile application [73]. Making API calls to Smapshot relies on having internet connection. For example, the following API request is called in the mobile application to reach an endpoint and fetch the resource of a photo from Smapshot:

<https://smapshot.heig-vd.ch/api/v1/data/collections/31/images/500/185747.jpg>

6.1.3 Dependencies

It is possible to add and download external binaries or libraries in Android Studio into a project, also known as dependencies [74]. By adding dependencies in the build system Gradle, Android Studio will download and make the dependencies available into the project. During the development of this mobile application, osmdroid, Volley, Picasso and HTextView was used.

Osmdroid

Osmdroid is a dependency used for displaying and interacting with OpenStreetMap data [75]. By adding osmdroid to the project, the mobile application can render and display a map for the users. The map is used to show locations of old photos as well as display the user's current location. The reason for choosing OpenStreetMap and osmdroid instead of other map services was because OpenStreetMap is free to use and has more permissive licensing.

Volley

Volley is a dependency that handles network operations in Android applications [76]. Volley is used for establishing a network connection and sends requests from the mobile device to get information. Volley was used in order to get information from Smapshot's API.

Picasso

Picasso is another dependency used for image downloading from websites, developed by Square Inc [77]. Picasso is open source and is used in the mobile application to display images fetched from Smapshot's API in order to show photos in the mobile application.

TextView

TextView is a dependency that supports animation effects to texts [78]. TextView was used to show animation effects to make the mobile application feel more alive and interactive to the user.

6.1.4 Github repository

The project use Github which is a cloud-based service that helps developers store and manage their projects. The codebase for this thesis work can be found here:

<https://github.com/Christoffer9612/smapshot-application>

6.1.5 Agile Software Development

The project use agile software development, an iterative development model where the main principle is to deliver software continuously [79]. Agile software development allows requirements of a product to change throughout the process which makes it a flexible methodology.

Scrum

Scrum is the framework used throughout the project and is often used in agile contexts for controlling and managing software [80]. It was designed to increase the speed of development.

6.2 Development process

The project used agile software development as a development method. In the work of this thesis project, the agile model was applied because it is iterative, which also is the case for a design process. Every week, a meeting was held with the supervisors from Switzerland. The purpose of these meetings were to update the supervisors on three topics: progress made, plan for the next week and discussion or questions. Every week was therefore seen as a sprint. Every second week, a meeting was held with the supervisors from Lund University to update them on the progress made and reviewing report writing.

Scrum was used as framework and complement to working with agile software development in the development process. It consisted of some principles of scrum to facilitate the progress. After every sprint, tasks were defined and created as stickers. These stickers were placed on a scrum board, created in Miro, seen in figure 6.3. The stickers were divided into four different categories: design, geo, development and other. Organizing tasks into different categories was used to be able to separate the different types of tasks. Alfred was responsible for completing design tasks, Christoffer was responsible for geo tasks and the rest were shared responsibility. A backlog was introduced, to track upcoming tasks and prioritize them from high to low where the most urgent ones had highest priority.

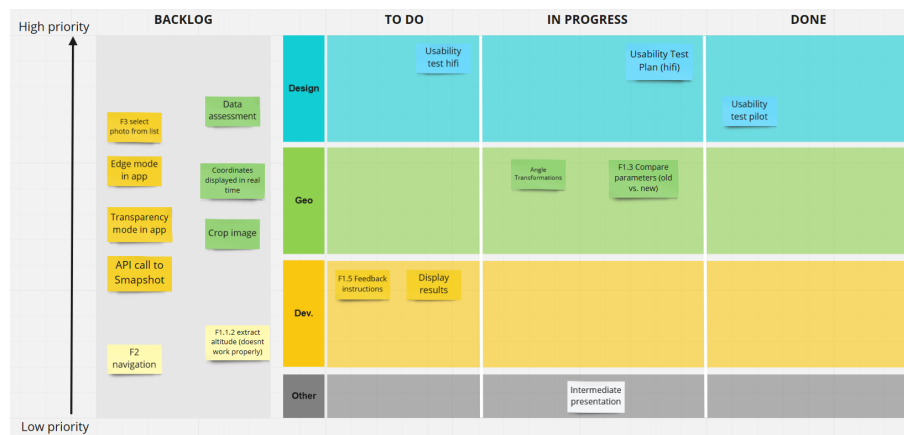


Figure 6.3: Scrum board

6.3 Implementation

6.3.1 Portrait vs. landscape mode

The mobile application is developed for when using the device in portrait mode (held vertically). Meaning that if the device is used in landscape mode, the orientation angles will not provide the correct orientation.

6.3.2 Compensating for magnetic declination

As mentioned, Android operating system calculates the azimuth of the device based on the magnetic North Pole. To compensate for the declination and to base the azimuth on True North, which is what the azimuth value from Smapshot is based on, the declination was calculated based on the location of the photograph, and added to the azimuth of the device. The magnetic declination in Android Studio is calculated with with the public class `GeomagneticField` and the `getDeclination`-method, with the parameters longitude, latitude and altitude as input [81].

6.3.3 Normalisation of orientation angles

Orientation angles that have been calculated by georeferencing the photographs with Smapshot, could range from negative values to values larger than 360° . The orientation angles for the device is returned as radians ($0-2\pi$). To able to compare them with the orientation angles for the historical photograph they firstly need to be converted to degrees. Furthermore, the orientation angles of both the mobile device and of the historical photograph from Smapshot were normalised. The azimuth values were normalised from 0° to 360° , while the tilt and the roll were normalised from -180 to 180° .

6.3.4 User instruction

The application has two techniques for guiding the user to recapture the photo. The first one is using the orientation angles. The mobile application computes the orientation angles in real time and displays them on the camera view. This serves as some guidance for the user. Furthermore, the real time orientation angles were made to turn green when in a span of ± 5 degrees of the desired orientation, which is the orientation for the camera that captured the historic photo.

For further instruction an arrow next to the azimuth orientation angle was implemented. This arrow illustrates to the user which way to turn to achieve the desired azimuth angle. The arrow either points left or right. As rotation clockwise from True north would increase the azimuth, and turning counterclockwise would decrease the azimuth. See Figure 6.4 for these features.

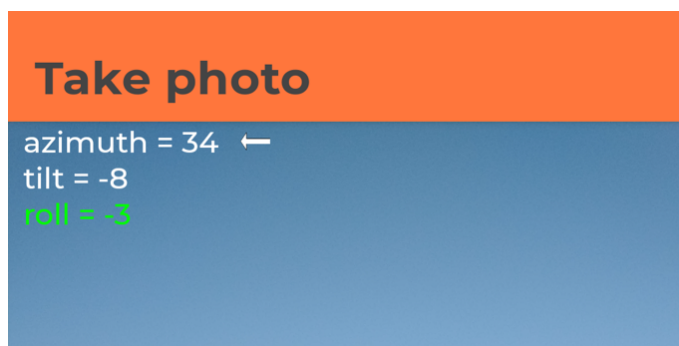


Figure 6.4: The orientation angles displayed in real time, turning green within a span of $\pm 5^\circ$, and an arrow illustration desired azimuth rotation

The transparency feature is the other way of instructing the user. The transparency feature is used as a visual guidance. This feature enables the historic photo to be placed on top of the camera view, with a slide that increase or decrease the transparency of the historic photo. The transparent historic photo can also be zoomed in and out. When using the transparency feature the user can use the transparent historic photograph as a way to align certain feature with the view of the camera. For instance, in the historic photograph used, the user can try to match the mountain tops, as well as the shoreline with the historic photograph. The transparency feature is illustrated in Figure 6.5.



Figure 6.5: The transparency feature, guiding the user to recapture the photograph

6.3.5 Comparing the orientation angles

To determine the outcome of the recaptured photograph the difference between the orientation angles of the historic photo and recaptured photo is calculated. The shortest rotation from the device orientation angle to the corresponding orientation angle of the historical photo is calculated. The mean difference of the three orientation angles is then calculated. Based on this mean, a grade system ranging from E to A, was implemented.

- **0° to 4°** - Grade A.
- **5° to 14°** - Grade B.
- **15° to 24°** - Grade C.
- **25° to 34°** - Grade D.
- **35° to 44°** - Grade E.
- **34° to 360°** - Grade F.

Chapter 7

Rephotography Data Quality Assessment

7.1 Test 1

There are two features that are used to guide the user to retake the photo. One being the orientation angles displayed on the camera view in real time, and the other being the transparency feature. The test compares the outcome of these two techniques.

The mobile application was used to recapture the two photographs illustrated in Figure 6.2. The user was guided to the location of the photo. The test-user was then asked to recapture the photo 8 times, by only using the real time orientation angles displayed on the camera as guidance. The test-user then recaptured the same picture another 8 times, but this time only using the transparency mode available in the application as guidance. The deviation for each of the three orientation angles, from the corresponding angle for the historic photo, were calculated. As well as the mean error and the standard deviation (SD). The historic photo was recaptured for this test by test-users T1, T2 and T3. Data regarding the test-users can be seen in Figure 7.1.

Table 7.1: The four test-users for the data quality tests

Test-user	Mobile experience (1-5)	Prior knowledge of app (1-5)
T1	5	1
T2	5	1
T3	5	5
T4	5	5

T1 performed the test under foggy weather conditions, making it more difficult to use the transparency feature, because of limited visibility. Recapturing a photo with limited visibility due to weather conditions is not of much interest, but the test illustrates the difficulties in using the transparency feature when there are no features in the view to align the transparent photo with. For instance, if a view used to have trees, that have now been cut down.

7.1.1 Historic photo results

The outcome of one of the recaptured historic photos, when using the real time orientation angles as guidance is shown in Figure 7.1.



Figure 7.1: The historic photo (left) and one of the recaptured photos (right) when using the real time orientation angles as guidance

The distribution of the deviation of azimuth, tilt and roll values, when using the real time orientation angles, for T3 is shown in Figure 7.2. The same graphs were calculated for all test-users and can be found in Appendix C. The mean error and standard deviation can be seen in Table 7.2 and Table 7.3

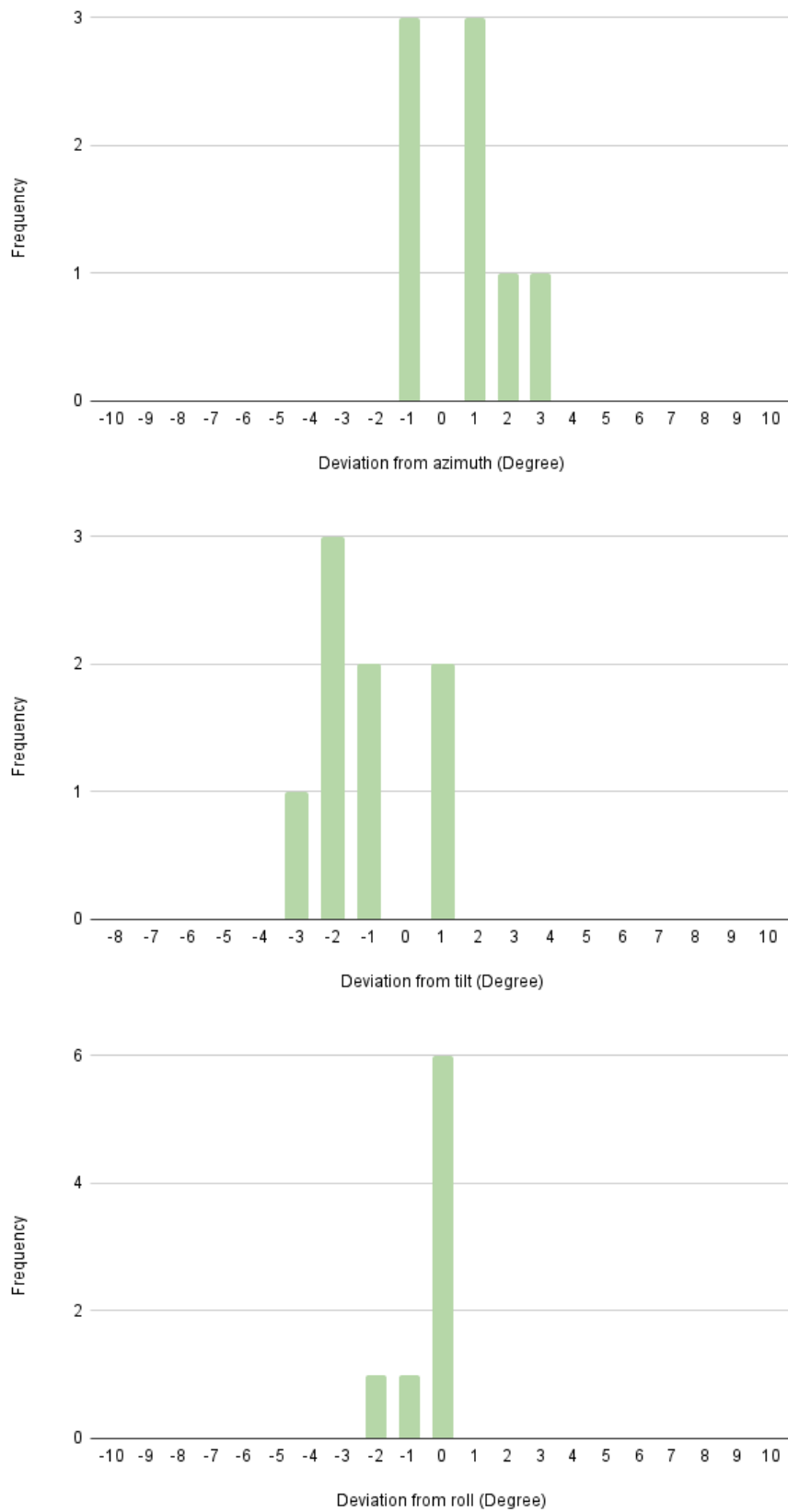


Figure 7.2: Deviation when using real time orientation angles as guidance when recapturing the historic photo (T3)

Table 7.2: Mean error and standard deviation for each of the three orientation angles, when using the real time orientation angles as guidance

Test User	Azimuth Mean Error °	Azimuth SD °	Tilt Mean Error °	Tilt SD °	Roll Mean Error °	Roll SD °
T1	3.4	1.85	2.8	2.04	1	1.10
T2	1.25	1.87	2.25	1.48	1.75	1.94
T3	1.38	1.41	2.13	2.34	0.38	0.70

Table 7.3: Mean error and standard deviation for each of the three orientation angles, when using the transparency feature as guidance

Test User	Azimuth Mean Error °	Azimuth SD °	Tilt Mean Error °	Tilt SD °	Roll Mean Error °	Roll SD °
T1	5.2	1.47	4.8	2.23	0.8	1.67
T2	4.75	2.44	1.25	1.56	1.25	1.20
T3	2	2.29	1.63	1.36	0.88	0.33

7.1.2 Test photo results

The outcome of the recaptured photos, when using the transparency feature as guidance by test-user T4 can be seen in Figure 7.3. Mean error and standard deviation can be seen in Table 7.4 and Table 7.5.



Figure 7.3: The historic photo (left) and one of the recaptured photo (right) using the transparency feature

Table 7.4: Mean error and standard deviation for each of the three orientation angles, when using the real time orientation angles as guidance

Test User	Azimuth Mean Error °	Azimuth SD °	Tilt Mean Error °	Tilt SD °	Roll Mean Error °	Roll SD °
T4	2.8	3.28	1.6	1.99	0.8	0.87

Table 7.5: Mean error and standard deviation for each of the three orientation angles, when using the transparency feature as guidance

Test User	Azimuth Mean Error °	Azimuth SD °	Tilt Mean Error °	Tilt SD °	Roll Mean Error °	Roll SD °
T4	4.5	2.84	1.6	1.11	1.9	1.1

7.2 Test 2

A photograph with already known location and orientation was provided. Since the location and orientation for the photo already is known, the photograph could serve as a "ground truth". This photograph was then uploaded and georeferenced using Smapshot, and then recaptured using the application. This gives us an idea of how large the errors of the georeferenced parameters are, as well as how they compare to the orientation of the rephotograph taken with the application.



Figure 7.4: The photo that served as a ground truth and was georeferenced with Smapshot

The known orientation angles of the photo in Figure 7.4

- **azimuth** = 150°
- **tilt** = 6°
- **roll** = 3°

And when georeference with Smapshot the results are:

- **azimuth** = 148°
- **tilt** = 5°
- **roll** = 3°

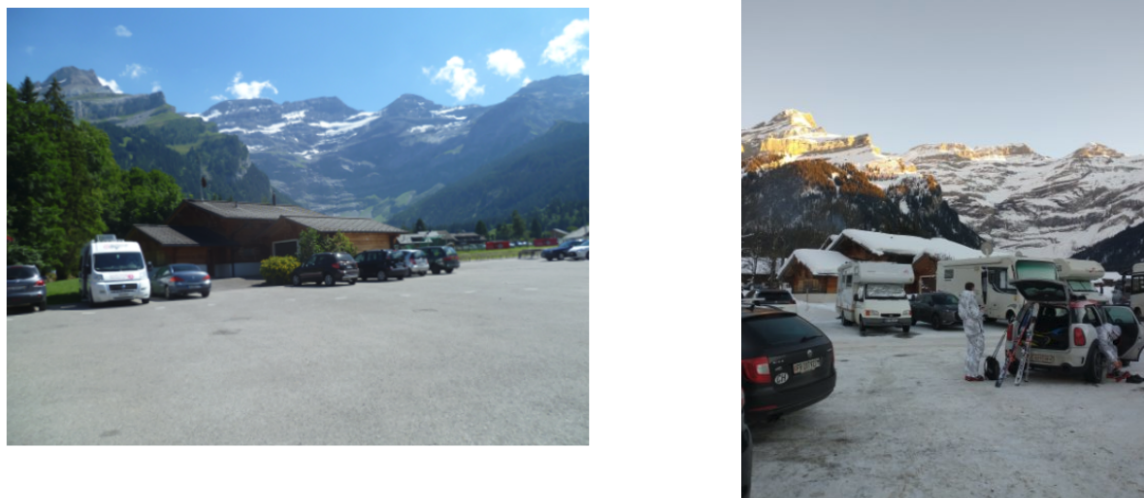


Figure 7.5: The recaptured photo (right) and the photo with known orientation and location (left)

The recaptured photo and the historic photo with known parameters is shown in Figure 7.5. The orientation angles for the recaptured photo, using the mobile application:

- **azimuth** = 148°
- **tilt** = 1°
- **roll** = 2°

7.3 Test 3

For this test, the test photo (Figure 6.2) and one of the recapture photos, were compared by identifying 6 corresponding points in the two photos and then using the pixel coordinates of the points as input in the software Gtrans [82]. Gtrans calculates the relative geometric transformation between the coordinate systems of the images. The photos consists of pixels that make up a Cartesian coordinate system. If (A, B) is a line segment between the pixels A and B , and T is the geometric transformation, then T transforms (A, B) into another straight line segment (A', B') , where $(A') = TA$, and $(B') = TB$ [83].

The transformation used in Gtrans is a 2D similarity transformation. A similarity transformation preserves angles. It translates, rotates and scales the coordinate system of the original photo to match the coordinate system of the recaptured photo. When the optimal match is calculated, the differences in pixel values are compared between the transformed photo and the original photo. Values regarding the differences in pixel values between the points, like the RMS for the x- and y-coordinates and the max radial error are calculated. The radial error is the distance between the corresponding points. The points that were identified and used as inputs into Gtrans, can be seen in Figure 7.6.



Figure 7.6: The points that were identified and used as inputs in Gtrans

The root mean square error (RMS) of the x- coordinates was **16.61px**, the RMS of the y- coordinates was **13.40px**, and the max radial error was for **point 2**, and was **25.25px**. The photos have a resolution size of 3000x4000 pixels. The pixel size of the photos is around 0.35mm. This means that a max radial error of 25.25px corresponds to $25.25 \times 0.35 = 8.84\text{mm}$. For a scale bar of pixels see Figure 7.7



Figure 7.7: Visualisation of how pixels correspond to distance in the photo

Chapter 8

Discussion

8.1 Mobile application

"How to use" instructions

According to both usability sessions, majority of test-users found the instructions on how to use the mobile application difficult to interpret correctly. One test-user suggested learn by doing instead since the mobile application is not too complicated and does not have too many different features to experiment with.

Involve users socially

Two test-users suggested having the mobile application connected with social media in various ways. Throughout the project, discussions regarding involving users socially was brought up a few times. Due to lack of time, this field was never researched and should be investigated more thoroughly for future considerations. As an example, involving users socially goes hand in hand with gamification and leaderboards.

8.1.1 Techniques for retaking photos

User perspective

According to all eight test-users from the usability tests, having transparency as a feature for retaking photos makes sense. On the other hand, the orientation angles were not perceived as clearly. In the first run of usability tests, no test-user noticed the orientation angles. On the second run, all three noticed the orientation angles and only one of three test-users understood what they were measuring. The explanation for this is that the orientation angles changed color from white to green in the second usability test which made them more attentive. In addition, it could also be that the orientation angles changed depending on how the phone was held because they were measured in real time which was not the case for the hi-fi design, where the angles were fixed and acted as a placeholder for visualization purposes only.

To mitigate the risk of not understanding the orientation angles, an explanation can be included for the orientation angles. Another solution mentioned from one test-user can be to replace the orientation angles with something more user friendly - a floating dot on the center of the camera view and having visual feedback for how the phone should be tilted and held. This was also brought up by the developer from Smapshot as a suggestion, but was never implemented due to lack of time and research. However, this should be considered and experimented with for future work of the mobile application.

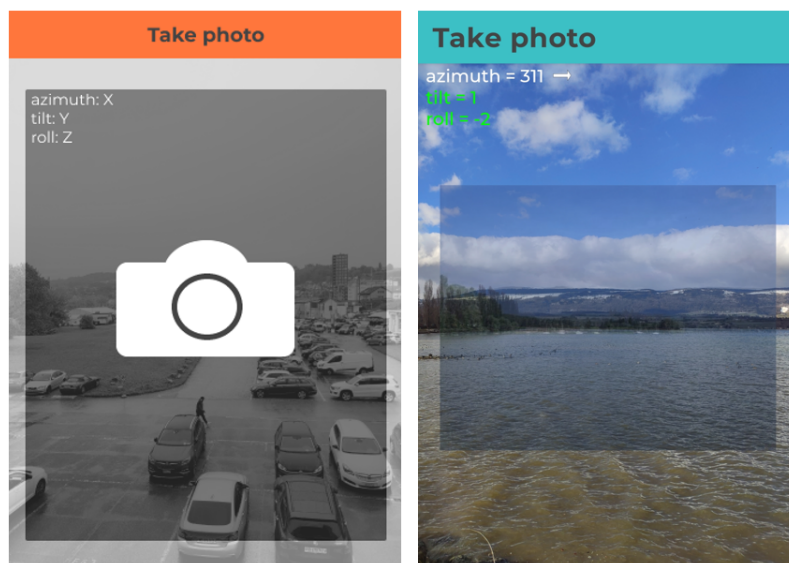


Figure 8.1: Camera view displaying orientation angles from the hi-fi prototype (left) and the final prototype (right)

Altering orientation after recapture

Another possible technique for recapturing photos is to alter the part of the photos orientation after the picture is taken, rather than communicating it to the user in real time. For instancing, rotating the picture according to the difference in roll angle values. For our application we experimented with providing instructions to the user after the photo was recaptured instead of doing it in real time. This proved to be difficult because the users tends to change the orientation of the phone once the photo was recaptured, making it difficult to understand instructions that were based on the orientation of the device at the exact moment of when the photo was recaptured.

8.1.2 Gamification

Early on in the process, it was decided to include some sort of gamification aspect to the mobile application to keep interest of usage, such as receiving points when recapturing a photo. From the first usability tests in the second iteration, all test-users felt that points motivated them to perform better. However, in the third iteration, no test-user felt that having grades ranging from A (best) to F (worst) motivated them to perform better. One reason for this can be because the hi-fi prototype included a leaderboard with fictional users which was not the case for the implemented prototype. In other words, the prototype was lacking a leaderboard comparing score against other made up users. Another reason can be the difference between having points and grades where points feels more like a game mechanic. One case study showed similar results between points and grading. A class of computer science students with gamified aspects of learning had higher attendance level and completed more homework per person than the non-gamified students [84]. The student group that were gamified used techniques like leaderboards and points meanwhile the rest had traditional academic grading.

In order to have users motivated, the points system should be implemented again but based on difference in orientation angles instead of based on accuracy. As mentioned earlier, there are sources of error from smartphone sensors and the georeferencing process in Smapshot which makes it impossible to determine accuracy. Furthermore, a leaderboard should also be included.

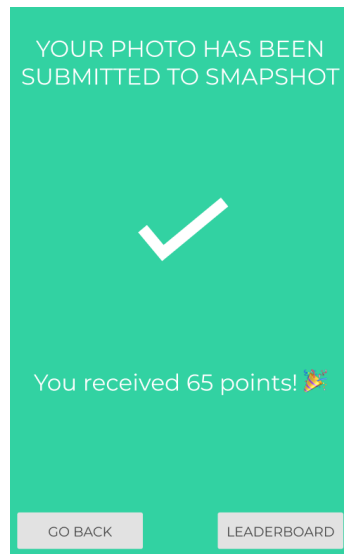


Figure 8.2: Receiving points from the hi-fi prototype

8.2 Design process

Survey results

The survey was answered by 38 Smapshot volunteers and 14 people from social networks. This means that the majority of survey responses was collected from Smapshot volunteers. In other words, the survey results are a bit biased towards Smapshot volunteers.

Personas

By sending out surveys, it was easy to capture data about different age groups and to characterize the different personas on a basic level. However, more research could be performed to get in-depth information about the different target groups by conducting interviews or having focus groups for further market research about user needs for the product.

The challenge of having personas was to recruit test-users for the usability tests that matched the persona profiles. For instance, none of the test-users resembled the second persona, David Berg - a male in his mid 40s, figure 5.6. This means that no feedback was provided by the second largest user group that had an interest in using the mobile application. What influenced the selection of test-users were several factors such as contacts available, social networks, language barriers (lack of English or Swedish) and the outbreak of covid-19 as mentioned in the third design iteration.

Usability testing

All of the 10 test-users that participated in the design process only attended one usability test each. The reason for this was to not form any additional bias towards the results and conclusions.

The reason for performing usability tests both remote and indoors was because it was easier for the host to observe, write notes and moderate the testing sessions simultaneously. Performing in a usability lab was discussed but excluded since the mobile application is not intended to be used inside a lab room. It was not possible to evaluate how different weather conditions affect the design of the mobile application by performing indoors. Another reason was the technical constraints, since the test phone did not have a SIM card and therefore no connection when outdoors.

Design vs. development

The biggest challenge was to balance the work between the design and development. By speeding up the design process in the first iteration, more room for development was created. It would be possible to stick merely with design and improving the current hi-fi design continuously, but a hi-fi prototype has its technical constraints. Without the developed prototype, it would not be possible to verify user feedback from the camera view and the orientation angles azimuth, tilt and roll. Furthermore, Christoffer's thesis work was based on having a simple, functioning prototype and therefore it seemed more sensible to contribute to the development of the prototype.

8.3 Rephotography data quality assessment

Test data set

It is important to be aware of the test data set, which is small for all three tests performed. This was due to difficulty of finding test user during a pandemic, as well as time limitations.

Camera differences

The application is developed for portrait mode (held vertically). There is difference in resolution size of photos taken in portrait mode, and photos taken in landscape mode. A photo taken in portrait mode is taller than it is wide. Taking photos in landscape mode results in a resolution size more similar to standard cameras. Also, the focal length of mobile cameras are typically short, which makes the field of view wider than that of standard cameras. Furthermore, the focal length distorts the photos at the edge of the photo.

Pixel comparison

For Test 3, the test photo, and the recaptured version of the test photo were both photographed with the same mobile device. This means that the camera calibration for is identical. It can therefore be assumed that the results would show larger differences in pixel values for a historic photo, taken with a camera with a different focal length.

The recaptured photo is not retaken from the exact location of the test photo. This means that the differences in pixel values for the corresponding points identified is larger in the foreground, than in the background. There are more points identified that are in the foreground, than in the background. These points will affect the results of the 2D similarity transformation, resulting in higher pixel differences between the transformed recaptured photo and the original photo.

Sensor accuracy

According to the literature study the compass uncertainty of smartphone devices is around 10° [27]. Since the azimuth is based on the compass value, it means that the azimuth uncertainty is around 10° . According to tests performed in this study, the azimuth uncertainty when using the application and the orientation angles as guidance, ranges from 2.8° to 3.4° . When using the transparency feature as guidance the azimuth uncertainty ranges from 2.0° to 5.2° . If the device sensors computes the azimuth with a 10° uncertainty, the difference between the results from using the orientation angles, and the transparency feature, would be larger. Meaning, the results from this study contradicts the results from the literature study, when it comes to the azimuth angle.

One reason for why the uncertainty seems to be smaller than indicated by the literature study is that the historic photos was recaptured in an open areas with no interference, like buildings, which could be one reason for why the uncertainty seem to be low. The test photo however, was taken from inside a building, which typically result in larger uncertainty to the

computed orientation angles. The compass was calibrated before performing the tests, this could further improve the results of the computed orientation angles.

Orientation angles vs transparency feature

Using the orientation angles as guidance when recapturing the photo results in a smaller differences of the orientation, than when using the transparency feature. But it is noticeable that the difference is not substantial. Because the result is based on the orientation computed by Smapshot, it means that the result is not based on parameters that are entirely accurate. This means that the results from using the transparency feature might more closely resemble the historic photo, even though the difference between the orientation angles of the historic photo and the recaptured one is larger. The required accuracy depends on what the image is to be used for. For more commercial purposes recapturing the photo with exactly the same orientation angles is perhaps not necessary. It could however be more important when using recaptured photos in photogrammetry practises.

Tilt and roll

The tilt, and especially the roll values are not as affected by the use of the transparency feature. This might be because both the tilt and roll values for photos tends to be similar for most photos. The camera device is usually held in a similar orientation when it comes to tilt and roll when taking photos.

The roll mean error is small, ranging from 0.38° to 1.75° , with a low standard deviation. A low standard deviation means a high precision. With the roll angle it is easier to achieve the desired orientation. The value is more static than the azimuth value.

Weather conditions

The mean errors of the azimuth and the tilt for test user T1 were higher compared to the other test users. Especially when using the transparency feature (5.2° azimuth mean error, 4.8° tilt mean error). T1 performed the test under bad weather conditions, making it harder to use the transparency feature as guidance, which resulted in greater differences in orientation between the historic photo and the recapture photos.

Georeferencing error

Test 2 was only performed with 1 photo, but it gives us an idea of the errors from the georeferencing process of Smapshot. The error ranged from 1° to 2° , with the larger error being to the azimuth angle. Recapturing the photo showed similar uncertainties as Test 1 did (4° tilt error, and 2° roll error).

8.4 Future work

Improvements to application

The current version of the mobile applications tracks the location of the device, and places the historic photos at their location on a map. However, the only navigation tool is the message, displayed on the camera view if the user is more than 20 meters away from the location, seen in figure 5.19. The navigation of the application can be improved.

Due to different camera calibrations, photos taken with older cameras and those taken with a mobile device will differ in field of view and resolution size. Implementing a feature that would allow the user to crop photos could be useful. Another improvement related to the visual difference between the recaptured photo and the historic photo could be to make the mobile application work in landscape mode.

The current version of the application uses the Smapshot API to fetch one photo. But the API could also be used to fetch the JSON files containing information about the photos. Integrating the Smapshot API to enable the recapturing of more than two photos in the current application would improve it. However, this is dependant on how many georeferenced Smapshot photos there are in the immediate area.

Artificial intelligence can be used to recognize photos taken in the same environments [85]. This could be used instead of having coordinates from a GPS and help users find their position. Also, there is research regarding the matching of geometric patterns in pictures [86]. It would theoretically be possible to have artificial intelligence help the user with the positioning of the camera view, but also with the information in the photos. Segments of artificial intelligence and machine learning is something that should be explored and researched upon in the future and how to involve it in a Smapshot mobile application.

Chapter 9

Conclusion

9.1 Geographic information conclusions

9.1.1. What kind of uncertainties can be expected when rephotographing a Smapshot photo (with known orientation) using a smartphone?

The computation of orientation angles with smartphone sensors results in a azimuth uncertainty of around 10° [27], a tilt and roll error of between 2.1° and 6.6° [27]. From test 2 we found a 2° azimuth error and 1° tilt error when georeferencing with Smapshot. Additionally there are uncertainties when using the mobile application, ranging from 2.0° to 5.2° for the azimuth angle, 1.5° to 4.8° for the tilt angle, and from 0.8° to 1.9° for the roll angle. From Test 3 we saw a RMS of 16.61px for the x-coordinates, and a RMS of 13.40px for the y-coordinates. The result of Test 3 was more affected by the test-user not standing on the exact location as the original photo, than by the orientation of the device.

9.1.2. How can a Smapshot mobile application be implemented for rephotography?

The app can be developed with the Smapshot API, that fetches photos and orientation data from the Smapshot platform. Open Street Map can be used for allowing the user to see current location and the location of the photo, as well as choosing which photo to recapture from the map. A Custom Camera API can be used to display the orientation angles in real time, turning to green from white when in a span close to the desired angles. Further, an arrow instructing the user which way to rotate to achieve correct azimuth angle can be used as guidance. A transparency feature can be implemented, allowing the user to use the historic photo and align it with the view of the camera. The results can be based on studying the difference between the orientation angles.

9.1.3. What problems have to be addressed in relation to the computation of the orientation angles of the device, and the comparison to the orientation angles of photos from Smapshot?

It is important to consider the angular convention so that it is the same as for the Smapshot system, in order to be able to compare the orientating angles. The magnetic declination needs to be considered and compensated for so that the system of orientation angles is based on True North. Both the orientation angles from Smapshot and the orientating angles of the device needs to be normalised in order to compare them.

9.2 Design conclusions

9.2.1. To what extent is there an interest of using a Smapshot mobile application?

According to the survey results, there is an interest of using a Smapshot mobile application. However, it is important to acknowledge the fact that the majority of survey responses are represented by Smapshot volunteers. The interest seems to be dominated by users intended to use the application occasionally (every two weeks or less). In addition, five out of eight test-users pointed out that the mobile application is better suited for travelling purposes.

9.2.2. Who are the potential end-users?

The potential end-users are represented by different age groups, seen from the survey results. The most common age group is 21 to 30 years old and the second most common is 41 to 50 years old. The majority of interest comes from men.

9.2.3. What are the challenges in its usage?

One challenge in using the mobile application is to understand the orientation angles. Suggestions has been made to replace the angles with something more user friendly. Another challenge is to understand what Smapshot is and how the mobile application works by reading the instructions. It should be experimented with a "learn-by-doing" approach instead.

9.2.4. How to keep interest of usage?

One way to keep interest of usage is by having a points system with a leaderboard when recapturing a photo. This resulted in motivating the test-users to perform better and therefore engaging the users. Another way to encourage usage is to involve users more socially and integrate social media with the application, allowing users to share and rate their work with each other.

References

- [1] M. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, 2007.
- [2] T. Osaragi and I. Niwa, "Development of system for real-time collection, sharing, and use of disaster information," *Mansourian A., Pilesjö P., Harrie L., van Lammeren R. (eds) Geospatial Technologies for All. AGILE 2018. Lecture Notes in Geoinformation and Cartography*, 2018.
- [3] J. Jedersberger, C. Stern, M. Wykstra, and G. Schaab, "Supporting the monitoring of cheetahs in kenya by mobile map apps," *Geospatial Technologies for All*, 2018.
- [4] Smapshot, "The participative time machine." <https://smapshot.heig-vd.ch/>, 2021 [accessed December 7, 2021].
- [5] O. S. Initiative, "The open source definition." <https://opensource.org/docs/osd>, 2006 [accessed October 11, 2021].
- [6] T. Produit and J. Ingensand, "Georeferencing of historical photos by volunteers," *The Annual International Conference on Geographic Information Science*, 2018.
- [7] S. Bae, A. Agarwala, and A. Agarwala, "Computational rephotography," *ACM Transactions on Graphics*, 2010.
- [8] Y.-B. Shi, F.-P. Tian, D. Miao, and W. Feng, "Fast and reliable computational rephotography on mobile device," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2018.
- [9] R. West, A. Halley, D. Gordon, J. O'Neil-Dunne, and R. Pless, "Collaborative rephotography," in *ACM SIGGRAPH 2013 Studio Talks*, pp. 1–1, 2013.
- [10] A. Gore, "An inconvenient truth." Paramount Classics, 2006.
- [11] F. C. Hall, "Photo point monitoring handbook," *Tech. rep. PNW-GTR- 526, USDA Forest Service.*, 2002.

- [12] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-centered design." Draft work.
- [13] J. Mao, K. Vredenburg, P. Smith, and T. Carey, "The state of user-centered design practice," *Communications of The ACM*, vol. 48, pp. 105–109, 3 2005.
- [14] T. Lowdermilk, *User-Centered Design*. O'Reilly Media, Inc., 2013.
- [15] Esri, "Coordinate systems, projections, and transformations." <https://pro.arcgis.com/en/pro-app/latest/help/mapping/properties/coordinate-systems-and-projections.htm>, 2021 [accessed October 4, 2021].
- [16] GISGeography, "Cylindrical projection: Mercator, transverse mercator and miller." <https://gisgeography.com/cylindrical-projection/>, 2021 [accessed January 5, 2022].
- [17] Cesium, "Webmercatorprojection." <https://cesium.com/learn/cesiumjs/ref-doc/WebMercatorProjection.html>, 2021 [accessed November 2, 2021].
- [18] N. G. I. Agency, "Implementation practice web mercator map projection," *National Geospatial-Intelligence Agency (NGA)*, 2014.
- [19] N. Routley, "The problem with our maps." <https://www.businessinsider.com/the-mercator-projection-distorts-countries-2017-6?r=US&IR=T>, 2021 [accessed January 5, 2022].
- [20] T. Kuhlmann, P. Garaizar, and U. Reips, "Smartphone sensor accuracy varies from device to device in mobile research: The case of spatial orientation," *Behavior Research Methods*, (2021), 22-33, 53(1), 2021.
- [21] Android, "Sensor overview." https://developer.android.com/guide/topics/sensors/sensors_overview, 2021 [accessed October 10, 2021].
- [22] Android, "Sensor types." <https://source.android.com/devices/sensors/sensor-types>, 2021 [accessed October 29, 2021].
- [23] Android, "Position sensor." https://developer.android.com/guide/topics/sensors/sensors_position, 2021 [accessed October 10, 2021].
- [24] Android, "Location." <https://developer.android.com/reference/android/location/Location>, 2021 [accessed November 11, 2021].
- [25] National Centres For Environmental Information, "Magnetic Declination." <https://www.ngdc.noaa.gov/geomag/declination.shtml>, 2021 [accessed November 11, 2021].
- [26] National Centres For Environmental Information, "Geomagnetic Calculators, Maps, Models and Software." <https://www.ngdc.noaa.gov/geomag/models.shtml>, 2021 [accessed November 25, 2021].
- [27] J. Blum, D. Greencorn, and J. Cooperstock, "Smartphone sensor reliability for augmented reality applications," *McGill University, Montreal, Quebec Canada*, 2012.

-
- [28] Techopedia, “Assisted GPS (A-GPS).” <https://www.techopedia.com/definition/24218/assisted-gps-a-gps>, 2011 [accessed November 30, 2021].
- [29] Lantmäteriet, “DGNSS.” <https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/gps-geodesi-och-swepos/GPS-och-satellitpositionering/Metoder-for-GNSS-matning/dgnss/>, 2021 [accessed January 1, 2022].
- [30] R. Chen, T. Chu, J. Liu, X. Li, Y. Chen, L. Chen, and W. Xu, “Dgnss-c: A differential solution for enhancing smartphone gnss performance,” *A differential solution for enhancing smartphone GNSS performance. 27th International Technical Meeting of the Satellite Division of the Institute of Navigation*, 2014.
- [31] W. Fraczek, “Mean Sea Level, GPS, and the Geoid.” <https://www.esri.com/news/arcuser/0703/geoid1of3.html>, 2003 [accessed January 1, 2022].
- [32] Android, “Location.” <https://developer.android.com/reference/android/location/Location>, 2021 [accessed January 1, 2022].
- [33] Wolfram Mathworld, “Rotation Matrix.” <https://mathworld.wolfram.com/RotationMatrix.html>, 2021 [accessed November 8, 2021].
- [34] Stochastically, “What’s the best 3D angular co-ordinate system for working with smartphone apps.” <https://math.stackexchange.com/q/381649>, 2017 [accessed December 7, 2021].
- [35] J. E. Greivenkamp, “Field guide to geometrical optics,” *Society of Photo Opticala*, 2004.
- [36] C. Bozzini, M. Conedera, and P. Krebs, “A new tool for obtaining cartographic georeferenced data from single oblique photos,” *23rd international CIPA symposium*, 2011.
- [37] T. Schenk, “Introduction to photogrammetry,” *Department of Civil and Environmental Engineering and Geodetic Science The Ohio State University*, 2005.
- [38] D. Strausz, “Application of photogrammetric techniques to the measurement of historic photographs,” *GEO 522 - Reconstructing Historic Landscapes*, 2001.
- [39] S. Kang and H.-Y. Shum, “A review of image-based rendering techniques,” *In Proceedings of the IEEE/SPIE Visual Communications and Image Processing Conference. 2–13*, 2002.
- [40] P. V. Oort, *Spatial data quality: from description to application*. PhD thesis, Wageningen University Research, 1 2006.
- [41] S. Servigne and T. Libourel, “Spatial data quality components, standards and metadata. spatial data quality: An introduction,” *International Scientific and Technical Encyclopedia*, 2006.
- [42] P. A. Zandbergen, “Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning,” 2009.
-

- [43] S. Clark and E. Levy, “Compared to specialized gps devices, how good are smartphones for measuring gps coordinate data?,” *Information Technology Services Branch, Calrecycle*, 2013.
- [44] K. Merry and P. Bettinger, “Smartphone gps accuracy study in an urban environment,” *Warnell School of Forestry and Natural Resources, University of Georgia, Athens, GA, United States of America*, 2019.
- [45] M. Jones and G. Marsden, *Mobile Interaction Design*. John Wiley Sons, Ltd, 2006.
- [46] T. Fullerton, *Game Design Workshop*. A K Peters/CRC Press, 2018.
- [47] C. Wilson, *Brainstorming and Beyond: A User-Centered Design Method*. Morgan Kaufmann, 2013.
- [48] J. Tidwell, *Designing Interfaces - Patterns for Effective Interaction Design*. O’Reilly Media, Inc., 2011.
- [49] T. Miaskiewicz and K. Kozar, “Personas and user-centered design: How can personas benefit product design processes?,” *Design Studies*, vol. 32, pp. 417–430, 9 2011.
- [50] Xtensio, “Create powerful business content together with Xtensio.” [://xtensio.com/](https://xtensio.com/), 2022 [accessed February 7, 2022].
- [51] W. C. Tomlin, *UX Optimization*. Apress, Berkeley, CA, 2018.
- [52] L. Takayama and J. Landay, “High-fidelity or low-fidelity, paper or computer choosing attributes when testing web prototypes,” *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 46, 9 2002.
- [53] Microsoft, “Microsoft Word.” <https://www.microsoft.com/sv-se/microsoft-365/word>, 2022 [accessed February 7, 2022].
- [54] G. Zichermann and C. Cunningham, *Gamification by Design - Implementing Game Mechanics in Web and Mobile Apps*. O’Reilly Media, Inc., 2011.
- [55] M. Gaigg, “Ux/ui best practices for designing map apps.” Esri User Conference Technical Workshops, San Diego, 7 2017.
- [56] Google Maps, “.” <https://www.google.com/maps>, 2021 [accessed December 5, 2021].
- [57] J. Rubin and D. Chisnell, *Handbook of Usability Testing*. Wiley Publishig, Inc., 2008.
- [58] M. Hammontree, P. Weiler, and N. Nayak, “Remote usability testing,” *Interactions*, vol. 1, pp. 21–25, 7 1994.
- [59] K. Thompson, E. Rozanski, and A. Haake, “Here, there, anywhere: remote usability testing that works,” *Proceedings of the 5th conference on Information technology education*, 2004.
- [60] T. Tullis, S. Fleischman, M. McNulty, C. Cianchette, and M. Bergel, “An empirical comparison of lab and remote usability testing of web sites,” *Usability Professionals Association Conference*, 2002.

-
- [61] Nielsen Norman Group, “Why You Only Need to Test with 5 Users.” <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>, 2022 [accessed February 7, 2022].
- [62] P. W. Jordan, B. Thomas, and B. W. Ian Lyall McClelland, *Usability Evaluation In Industry*. CRC Press, 1996.
- [63] J. R. Lewis and J. Sauro, “Item benchmarks for the system usability scale.,” *Journal of Usability Studies*, vol. 13, no. 3, 2018.
- [64] Microsoft, “Microsoft PowerPoint.” <https://www.microsoft.com/sv-se/microsoft-365/powerpoint>, 2022 [accessed February 7, 2022].
- [65] Zoom Video Communications, “Zoom.” <https://zoom.us/>, 2022 [accessed February 7, 2022].
- [66] Figma, “Figma: the collaborative interface design tool.” <https://www.figma.com/>, 2022 [accessed February 7, 2022].
- [67] T. Falk and R. Abler, “Intercommunications, distance, and geographical theory,” *Geografiska Annaler: Series B, Human Geography*, no. 62, pp. 59–67, 1980.
- [68] RunKeeper, “Track your Run - ASICS Runkeeper.” <https://runkeeper.com/cms>, 2022 [accessed January 7, 2022].
- [69] Android, “Android Studio.” <https://developer.android.com/studio>, 2021 [accessed December 8, 2021].
- [70] W3School, “JSON Introduction.” https://www.w3schools.com/js/js_json_intro.asp, 2021 [accessed December 12, 2021].
- [71] M. Meng, S. Steinhardt, and A. Schubert, “Application programming interface documentation: What do software developers want?,” *Journal of Technical Writing and Communication*, no. 48, pp. 295–330, 2018.
- [72] Android, “Camera API.” <https://developer.android.com/guide/topics/media/camera>, 2021 [accessed December 13, 2021].
- [73] Smapshot API, “Smapshot - The participative time machine.” <https://smapshot.heig-vd.ch/api/v1/docs/>, 2021 [accessed December 13, 2021].
- [74] Android, “Add build dependencies.” <https://developer.android.com/studio/build/dependencies#groovy>, 2021 [accessed December 6, 2021].
- [75] Osmdroid, “Osmdroid - OpenStreetMap Wiki.” <https://wiki.openstreetmap.org/wiki/Osmdroid>, 2022 [accessed January 7, 2022].
- [76] Android, “Volley overview.” <https://developer.android.com/training/volley>, 2021 [accessed December 6, 2021].
- [77] Picasso, “A powerful image downloading and caching library for Android.” <https://square.github.io/picasso/>, 2021 [accessed December 6, 2021].
-

- [78] Hanks-zyh, “Animation effects to text, not really textview.” <https://github.com/hanks-zyh/HTextView>, 2021 [accessed December 13, 2021].
- [79] S. Balaji and M. Murugaiyan Sundararajan, “Waterfall vs. v-model vs. agile: A comparative study on sdlc,” *International Journal of Information Technology and Business Management*, no. 1, pp. 26–29, 2012.
- [80] A. Srivastava, S. Bhardwaj, and S. Saraswat, “Scrum model for agile methodology,” in *2017 International Conference on Computing, Communication and Automation*, 2017.
- [81] Android, “Geomagneticfield.” <https://developer.android.com/reference/android/hardware/GeomagneticField>, 2021 [accessed November 11, 2021].
- [82] Lantmateriet, “Gtrans.” <https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/gps-geodesi-och-swepos/Transformationer/gtrans/>, 2022 [accessed February 2, 2022].
- [83] J. Foley and A. van Dam, *2D Geometrical Transformations*. Addison-Wesley Professional, 1994.
- [84] M. Laskowski and M. Badurowicz, “Gamification in higher education: A case study,” in *Management, Knowledge and Learning (MakeLearn) International Conference 2014 Human Capital without Borders: Knowledge and Learning for Quality of Life*, 2014.
- [85] D. Castro and J. New, “The promise of artificial intelligence,” *Center for Data Innovation*, vol. 115, no. 10, pp. 32–35, 2016.
- [86] A. De Bruyn, V. Viswanathan, Y. S. Beh, J. K.-U. Brock, and F. von Wangenheim, “Artificial intelligence and marketing: Pitfalls and opportunities,” *Journal of Interactive Marketing*, vol. 51, pp. 91–105, 2020.

Appendices

Appendix A

Survey

1. What is your gender?

- A. Male
- B. Female
- C. Other

2. What is your age?

- A. 0-20
- B. 21-30
- C. 31-40
- D. 41-50
- E. 51-60
- F. 61-70
- G. 71+

3. Where do you live?

- A. City
- B. Village
- C. Rural

4. What is your status?

- A. Student
 - B. Working
 - C. Retired
 - D. Other
-

5. Would you be interested in using a mobile application that allows you to re-take historical photographs of cities and landscapes?

- A. Yes
- B. No
- C. Maybe

5.1 If not or maybe - why?

6. How experienced are you in using mobile applications?

1 (little) - 10 (much)

7. How experienced are you in using navigation/map applications (Google Maps, etc.)?

1 (novice) - 3 (expert)

8. Would you be interested in participating in an interview with me?

- A. Yes
- B. No

Appendix B

Usability Test Plan

B.1 Second iteration

Tasks

A1) Find and select a photo to retake.

B1) Improve the accuracy of your newly taken photo by retaking it.

C1) Use the pre-designed route for capturing photos nearby ("Challenge of the day").

System Usability Scale Questionnaire

Question	Answer
1. I think the app was easy to use.	1-5
2. I think it was easy to understand and learn how the app works.	1-5
3. I think the different features in the app work well together (how to use, find photos, challenge of the day, navigation, scoreboard, etc).	1-5
4. I think the app was unnecessarily complicated.	1-5
5. I felt confident in what I was doing when I used the app.	1-5
6. I think I would use the app regularly.	1-5
7. I think I would use the app occasionally.	1-5
8. The points I received from submitting a photo motivated me to perform better.	1-5
9. It was easy to find and select old photos to retake.	1-5
10. I think the "challenge of the day" makes sense.	1-5
11. I think cropping mode makes sense when capturing a photo.	1-5
12. I think transparency mode makes sense when capturing a photo.	1-5

Interview

1. What do you think of the application and its current features?
2. Do you think the application worked equally well when finding photos to retake and using challenge of the day?
3. Would you consider using Snapshot mobile application in the future?
4. What improvements do you think need to be made?

B.2 Third iteration

Tasks

- A2) Understand what Snapshot is and how to use the app.
- B2) Retake the closest photo.
- C2) Try to improve your grade by retaking the same photo again.

System Usability Scale Questionnaire

Question	Answer
1. I think the app was easy to use.	1-5
2. I think it was easy to understand and learn how the app works.	1-5
3. I think the different features in the app work well together (how to use, find photos, select photo, navigation, grading, etc).	1-5
4. I felt confident in what I was doing when I used the app.	1-5
5. I think I would use the app regularly (more than once every week).	1-5
6. I think I would use the app occasionally (every two weeks or less)..	1-5
7. The grade I received from retaking a photo motivated me to perform better.	1-5
8. It was easy to find and select old photos to retake.	1-5
9. The display of a map when selecting a photo made sense.	1-5
10. I think transparency mode makes sense when capturing a photo.	1-5
11. I understood what the parameters azimuth, tilt and roll was about when opening the camera view.	1-5

Interview

1. What do you think of the application and its current features?
2. Would you consider using Smapshot mobile application in the future? Why or why not?
3. What did you think of the support that existed to retake a photo?
4. What improvements do you think need to be made?

Appendix C

Rephotography data quality tests

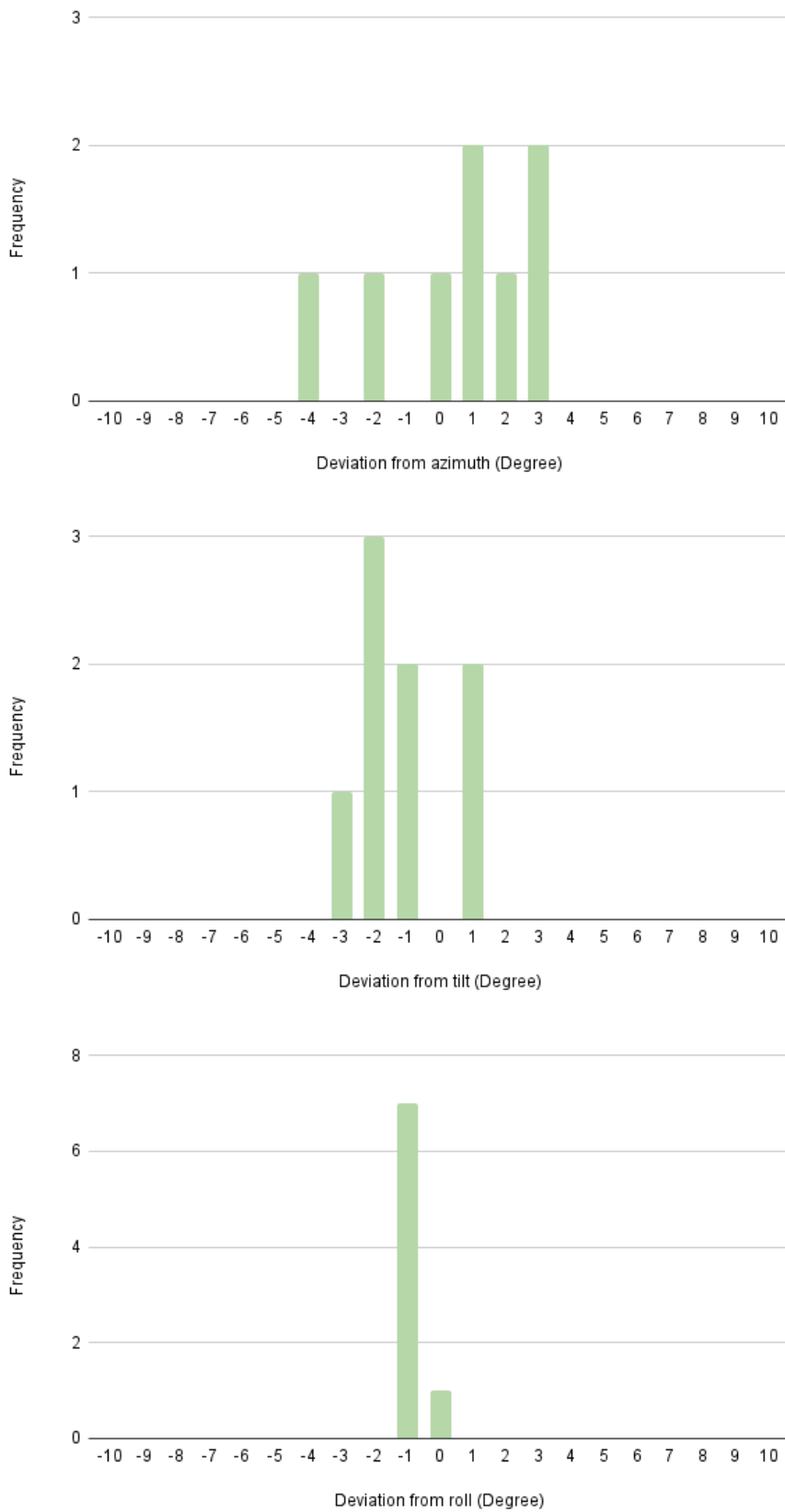


Figure C.1: T3, Deviation from orientation of historic photo when using the transparency feature as guidance when recapturing the historic photo

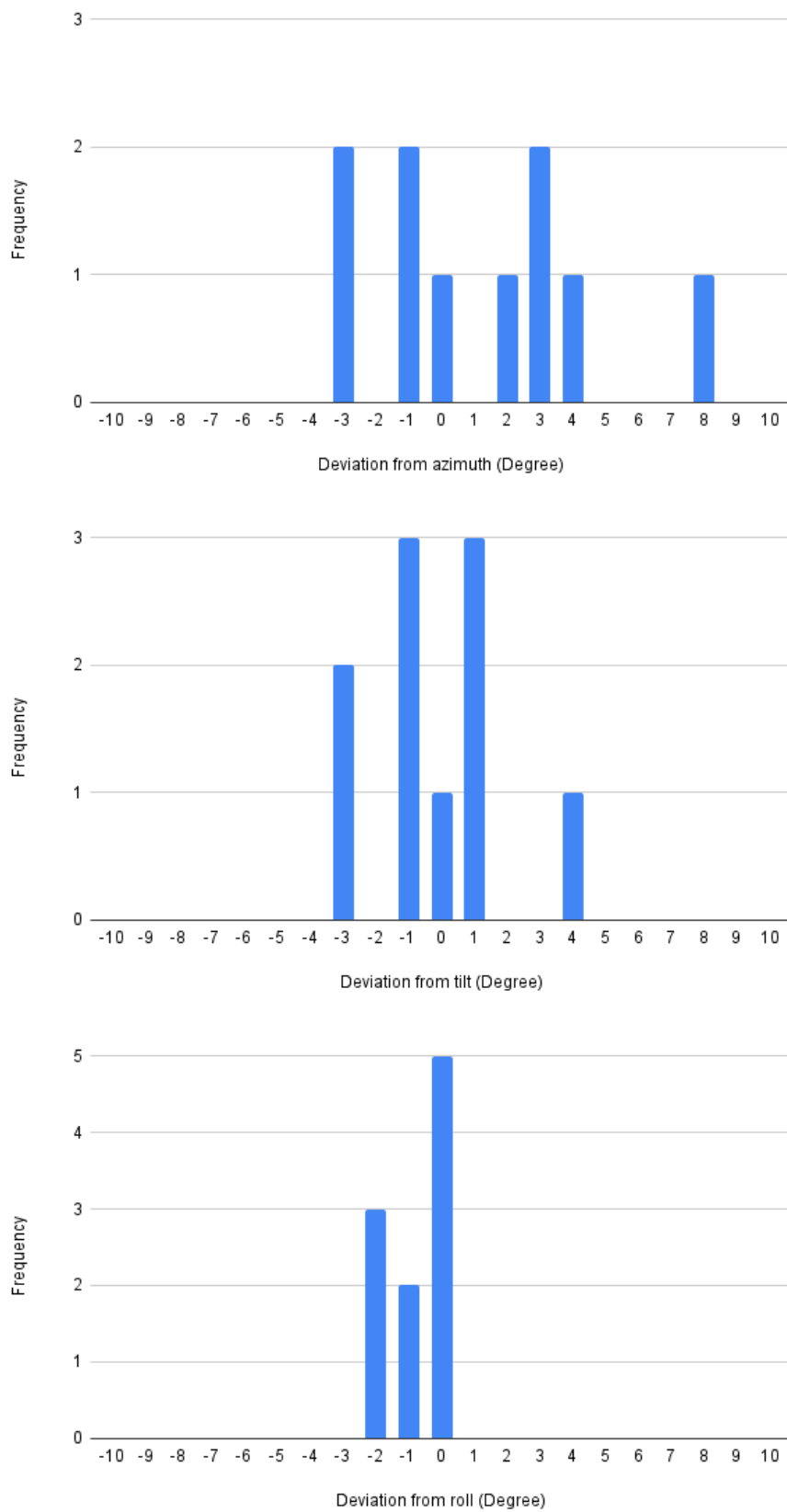


Figure C.2: T4, Deviation from orientation of historic photo when using real time orientation angles as guidance when recapturing the test photo

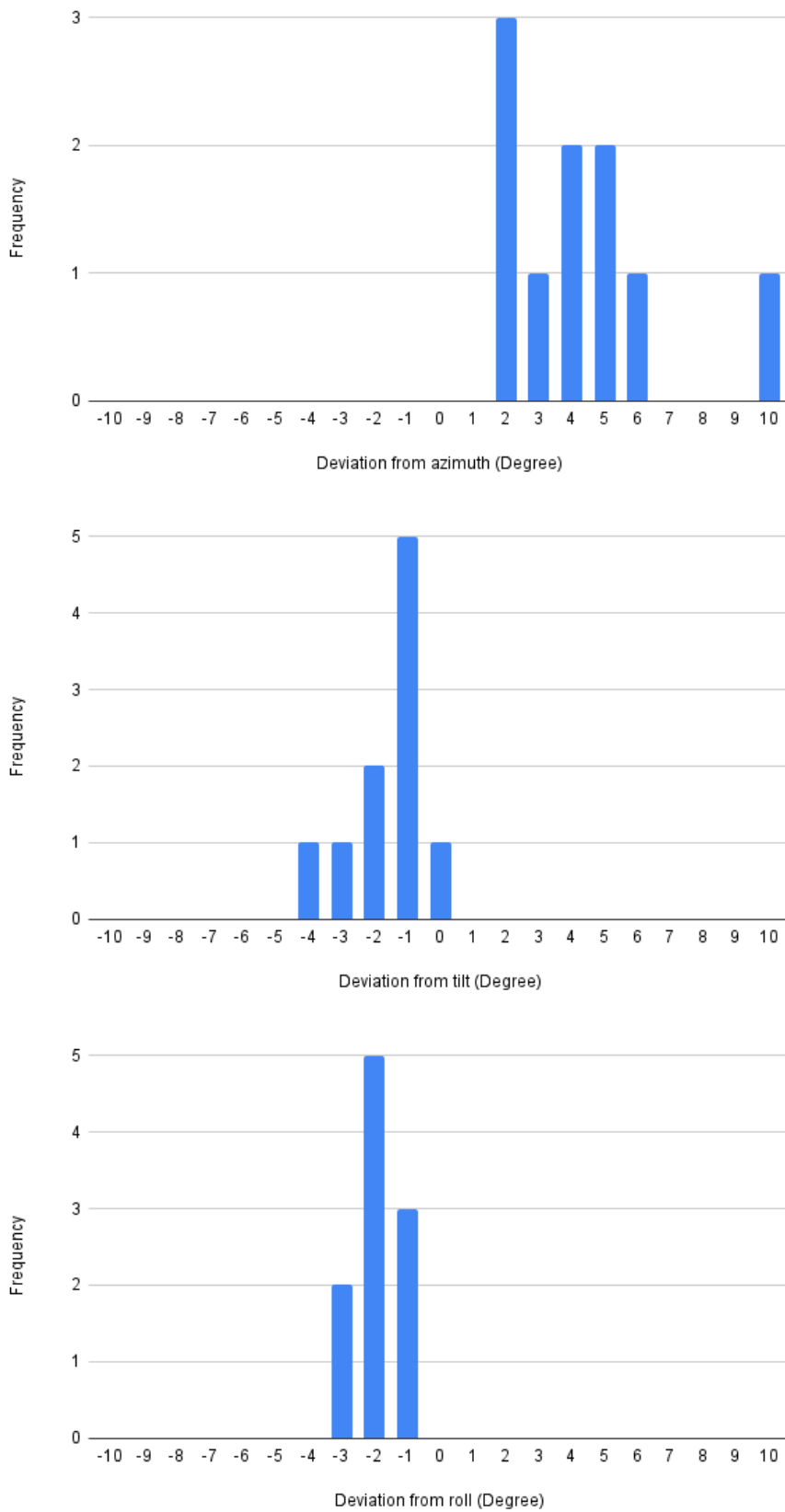


Figure C.3: T4, Deviation from orientation of historic photo when using the transparency feature as guidance when recapturing the test photo

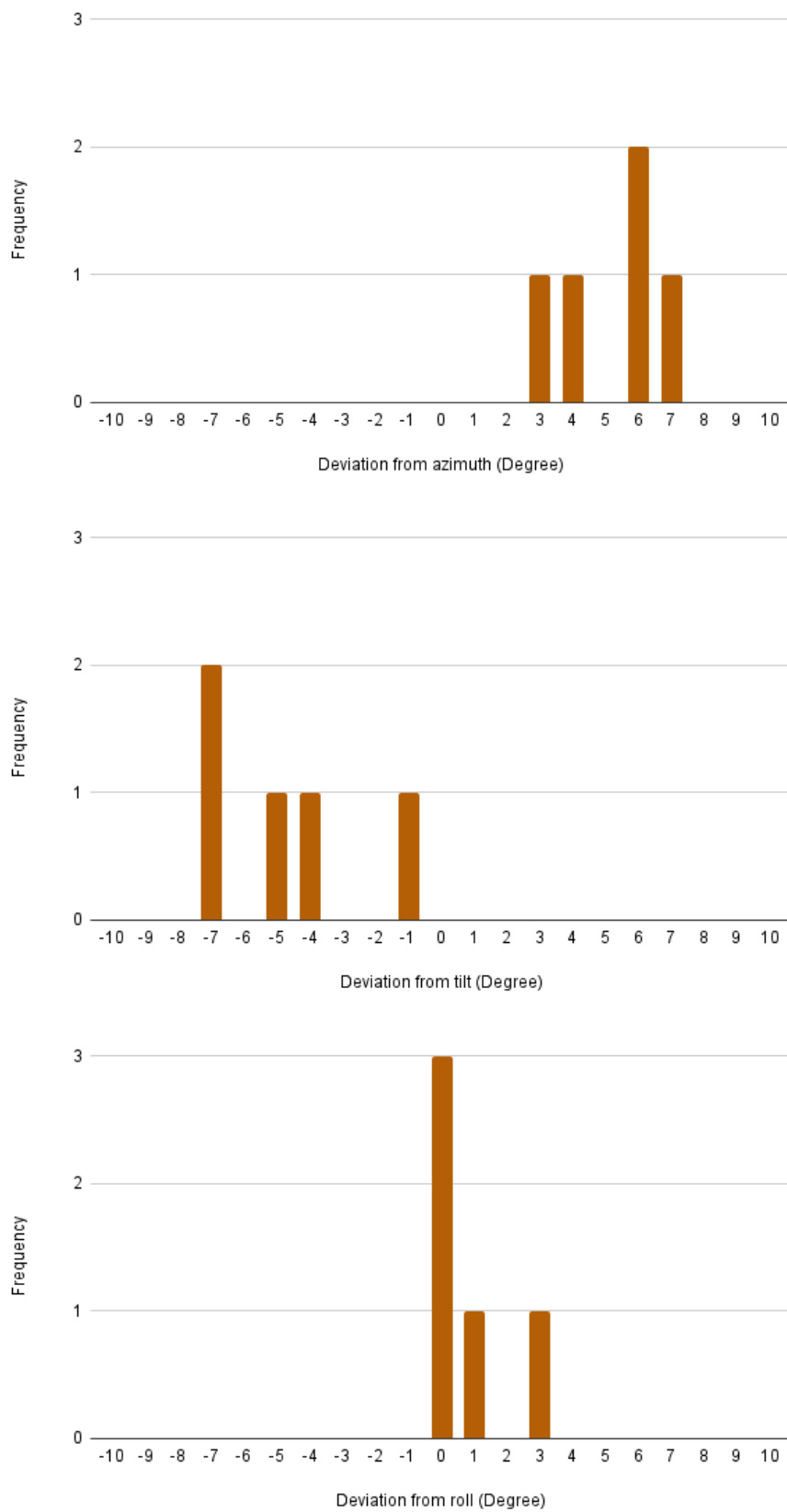


Figure C.4: T1, Deviation from orientation of historic photo when using transparency feature as guidance when recapturing the historic photo under bad weather conditions

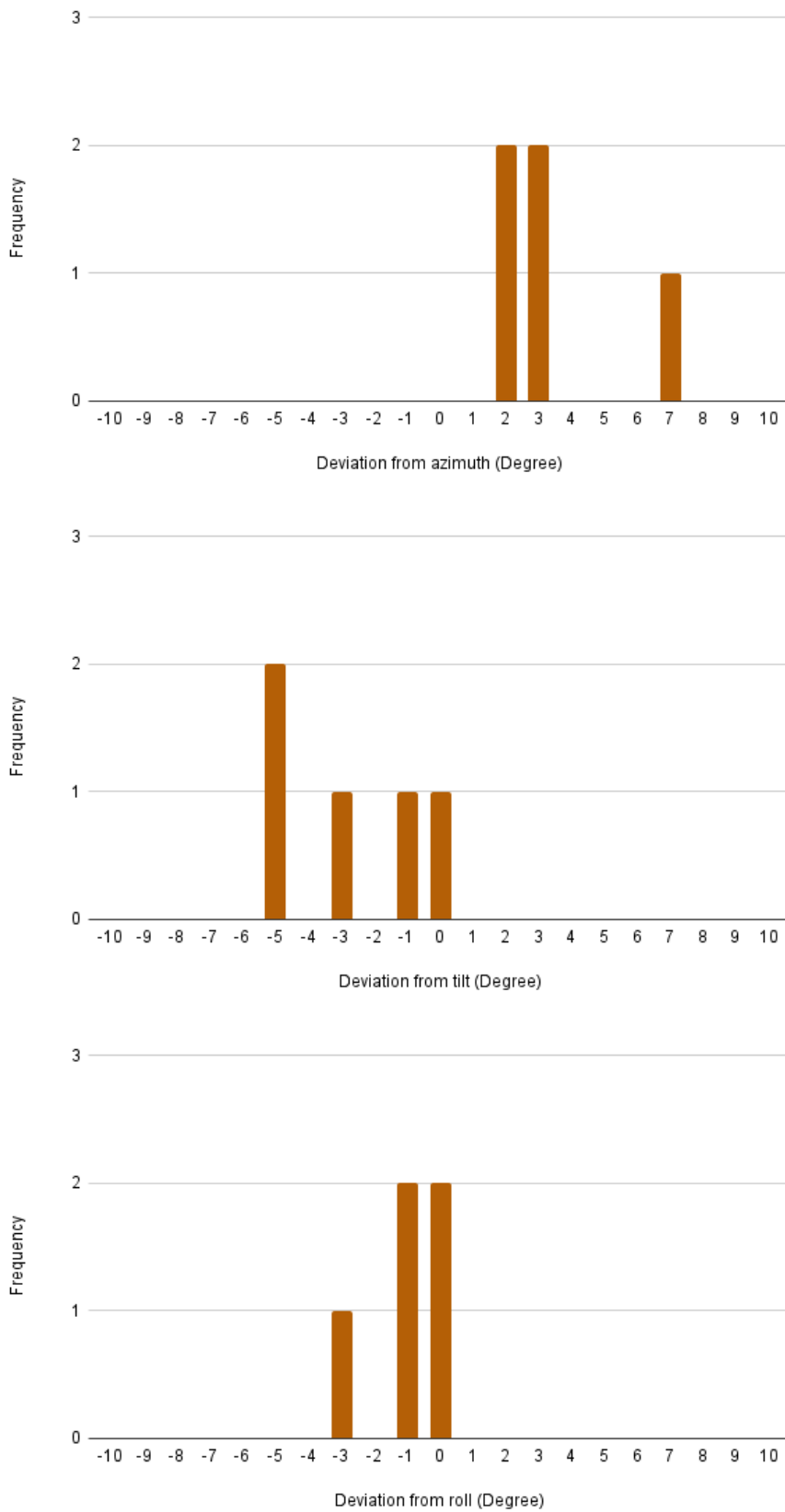


Figure C.5: T1, Deviation from orientation of historic photo when using real time orientation angles as guidance when recapturing the historic photo under bad weather conditions