

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Augmented Reality Solution for the Côa Valley Rock Art

Tomás Costa Fontes



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. António Coelho

July 25, 2023

Augmented Reality Solution for the Côa Valley Rock Art

Tomás Costa Fontes

Mestrado em Engenharia Informática e Computação

July 25, 2023

Resumo

A Realidade Aumentada (AR) é uma tecnologia que combina ambientes reais e conteúdo digital de uma forma interativa. Enquanto tecnologia, a AR tem crescido constantemente, sendo usada atualmente em várias áreas e aplicações com resultados muito positivos. Com o desenvolvimento das plataformas de software e hardware, a AR tornou-se acessível em várias formas, desde dispositivos móveis a dispositivos wearable, proporcionando experiências diversas aos seus utilizadores.

Apesar de a sua aplicação à área do turismo e cultura ser um tópico pouco explorado, apresenta grande potencial na melhoria das experiências culturais dos turistas. Alguns estudos sobre este tópico demonstraram a utilidade da AR na área, maioritariamente em ambientes de museu. Apesar disso, existe ainda pouca pesquisa sobre a aplicação de AR a gravuras de rocha plana, uma área que apresenta desafios adicionais quando comparada a ambientes de museu. Determinar o tipo de display e método de tracking a usar, lidar com as condições de ambientes exteriores e com as restrições físicas presentes nos locais de arte rupestre são os principais problemas nesta área.

O Vale do Côa contém mais de mil gravuras rupestres, espalhadas por mais de 80 locais que o público pode visitar. Em conjunto com a empresa InfoPortugal, esta dissertação tem como objetivo explorar como a AR pode ajudar a melhorar a experiência cultural dos visitantes das gravuras rupestres do Vale do Côa, como esta tecnologia pode ser aplicada a gravuras em rocha plana com sucesso, e quão recetivos estão os utilizadores a aplicações móveis de AR em visitas culturais.

O resultado desta dissertação deu origem a um protótipo de uma aplicação móvel de AR que ajuda os visitantes do Vale do Côa a visualizar gravuras de uma forma intuitiva e simples usando os seus dispositivos móveis. A aplicação delinea as gravuras contidas na área capturada pela câmara do dispositivo e mostra informação adicional acerca da arte rupestre visualizada. A informação adicional surge na forma de texto, imagens, ou mesmo animações posicionadas em cima de gravuras específicas para ajudar os utilizadores a compreender o contexto e história por detrás de cada gravura.

Abstract

AR is a technology that combines real-world environments and digital content in an interactive way. As a technology, AR has constantly been growing, currently used in various areas and applications with very positive results. With the development of software and hardware platforms, AR became accessible in many different ways, ranging from wearable to mobile devices, providing users with various experiences.

Although its application to the area of tourism and culture is not a widely explored topic, it presents great potential in improving cultural experiences for tourists. Some studies about this topic have demonstrated AR's usefulness in the area, mostly in museum settings. Still, there is little research on the application of AR to flat rock engravings, which presents additional challenges compared to museum environments. Figuring out what type of display and tracking method to use, handling outdoor conditions, and dealing with physical restrictions on rock art sites are the main problems in this area.

The Côa Valley contains over a thousand ancient rock engravings, scattered in more than 80 locations that the public can visit. In conjunction with the company InfoPortugal, this dissertation aims to explore how AR can help improve the cultural experience of the Côa Valley rock engravings visitors, how this technology can be successfully applied to flat rock engravings, and how receptive users are to AR mobile applications in cultural visits.

The result of this work is a prototype of a mobile AR application that aids the Côa Valley visitors in visualizing rock engravings seamlessly and straightforwardly using their mobile devices. The application outlines the engravings contained in the area captured by the device's camera and presents additional digital information about the captured rock art. Additional information is available in the form of text, images, or even animations placed on top of specific engravings to help users understand the context and story behind each drawing.

Acknowledgements

To my parents, for their unconditional support and love, for guiding me through my toughest times and for shaping me into who I am today.

To my sister, for going along with my craziness and being a source of inspiration.

To my grandparents, for all their wise advice and for always putting a smile on my face.

To Pinto, for being my piano, my partner during this 5-year adventure and for always having a way to make me laugh and enjoy life.

To Grande Pedro, for being a beacon of joy, the most pure and genuine friend I could ask for.

To Pedro, Baptiste, Johanna, Merle, Isabella, Ana, Sofia, Beatriz, Daniel, Paulo, Yasemin, Valentine and Adriana for all the unforgettable memories we created together, and for all the raclette nights that gave me the motivation to continue working on this project.

To Alexandre, Diogo, Ana, and all the InfoPortugal staff that helped this project come to life.

To Thierry and Patrícia, for their help during the visits to the Côa Valley rock art sites.

To Professor António Coelho, for guiding me through this entire process.

Tomás Costa Fontes

“Everything is experience for the future.”

Peter Sagan

Contents

1	Introduction	1
1.1	Context and Motivation	1
1.2	Research Questions	2
1.3	Objectives	2
1.4	Document Structure	2
2	Bibliographic Review	4
2.1	Introduction	4
2.2	Augmented Reality	4
2.2.1	Definition and History	4
2.2.2	Types of AR	7
2.2.3	Areas of Application	8
2.2.4	Augmented Reality Systems	9
2.2.5	AR Hardware	9
2.2.6	AR Software	10
2.3	AR Applied to Tourism	11
2.3.1	Mobile AR Applied to Cultural Visits	12
2.4	Related Work	13
2.4.1	Pose Estimation: Marker-based or Markerless	14
2.4.2	Handling Outdoors Conditions	15
2.4.3	Detecting Rock Engravings at Different Distances and Angles	18
2.4.4	Examples	19
2.5	Conclusions	19
3	Architecture and Specification	21
3.1	General Description	21
3.2	The Penascosa Rock Art Site	21
3.3	Application Requirements	22
3.3.1	Non-Functional Requirements	22
3.3.2	Functional Requirements	23
3.4	Architecture	24
3.5	User Experience	25
3.5.1	User Interfaces	25
4	Methodology/Solution	31
4.1	Tools and Technologies	31
4.1.1	Development Environment	31
4.1.2	AR Framework	31

4.1.3	Image Editing	33
4.2	UI Design	35
4.2.1	Initial Page	36
4.2.2	Engraving Detection Page	36
4.2.3	Additional Information Page	38
4.2.4	3D Visualization Page	38
4.3	Image Detection and Tracking	39
4.3.1	Approach	40
4.4	3D Visualization	43
4.5	Implementation Details	44
4.5.1	Virtual Outlines	44
4.5.2	Rock Information	44
4.5.3	2D Animations	45
5	Performance Evaluation	47
5.1	Testing	47
5.1.1	On-site Testing	47
5.1.2	Simulated Testing	47
5.2	Performance Evaluation - Image Detection and Tracking Improvements	48
5.2.1	Multiple reference images for each engraving/rock	49
5.2.2	Removing the background	52
5.2.3	Preprocessing the camera input	55
5.2.4	Using outlines as reference images	58
5.2.5	Phone flashlight	61
6	Usability Evaluation	62
6.1	Setup	62
6.2	Session Structure	62
6.3	Tasks	63
6.3.1	Task Scenario	63
6.3.2	Task A - Visualizing a 2D animation	64
6.3.3	Task B - Visualizing and controlling a 3D model	65
6.3.4	General Questions	66
6.4	Results	66
6.4.1	Participants Population	66
6.4.2	Task A - Visualizing a 2D animation	67
6.4.3	Task B - Visualizing and controlling a 3D model	73
6.4.4	General Questions	79
6.4.5	System Usability Scale	83
7	Conclusions	86
7.1	Improvements	87
7.2	Future Work	87
	References	89

List of Figures

2.1	The KARMA prototype	5
2.2	ARQuake game	6
2.3	ARTennis application on a Nokia device	6
2.4	Examples of markers used in marker-based methods	8
2.5	ToARist MAR application	12
2.6	AR application for the Pavia University History Museum	13
2.7	Virtual athletes running in the Olympia stadium	14
2.8	NFT system	15
3.1	Rock number 6 of the Penascosa rock art site	22
3.2	Application Architecture	24
3.3	Example of an application flow where the image tracking module keeps its state	25
3.4	Rock engravings without (a) and with (b) virtual outlines	26
	(a) Rock engravings without virtual outlines	26
	(b) Rock engravings with virtual outlines	26
3.5	Engraving selection interface	26
3.6	Example of displaying text in the AR scene, next to the engraving	27
3.7	Displaying information about an engraving on a different page	28
3.8	Slider that allows the user to control the scale of the 3D model: real-size model (a); bigger model (b)	29
	(a) Scale slider showing a 3D model at its real size	29
	(b) Scale slider showing a 3D model at a bigger size than its real one	29
3.9	Predefined scales for the 3D model	29
3.10	Slider that allows the user to control the rotation of the 3D model: front-facing model (a); side-facing model (b)	30
	(a) Rotation slider showing a 3D model facing the camera	30
	(b) Rotation slider showing a 3D model at a side angle	30
4.1	Reference images used in the initial prototype	32
	(a)	32
	(b)	32
4.2	Original engraving (a) and outline extracted from it (b)	34
	(a) Original rock engraving	34
	(b) Outline extracted from the engraving	34
4.3	Reference images artificially generated from an original picture	35
	(a) Original reference image	35
	(b) High brightness reference image	35
	(c) Low brightness reference image	35

(d) High contrast reference image	35
(e) Low contrast reference image	35
4.4 User Interface for the initial page	36
4.5 User Interface for the Engraving Detection Page before any engraving is detected	37
4.6 User Interface for the Engraving Detection Page after engravings having been detected	37
4.7 User Interface for the Engraving Detection Page after a specific engraving has been selected	38
4.8 User Interface for the Additional Information Page	38
4.9 User Interface for the 3D Visualization Page before planes having been detected	39
4.10 User Interface for the 3D Visualization Page after planes having been detected	39
4.11 Image detection and tracking process	40
4.12 Target rock captured only partially (a) and the engravings detected (b)	41
(a) Partial view of the target rock	41
(b) Detected engravings in the partial view of the rock	41
4.13 Reference images containing the entire rock, at different angles	42
(a) Reference image containing the entire rock from a front view	42
(b) Reference image containing the entire rock from a leftside view	42
4.14 Image containing the engraving to be used as reference	43
4.15 Colored virtual outlines	44
4.16 Outline texture for the horse engraving	45
4.17 Outline texture for the horse engraving with texture coordinates	46
5.1 Reference images from different angles: front (a), right (b), left (c)	49
(a) Frontal view image	49
(b) Right-side view image	49
(c) Left-side view image	49
5.2 Different viewing angles using the same outlines	51
(a) Frontal view engraving image with a virtual outline	51
(b) Right-side view engraving image with the frontal view virtual outline	51
5.3 Reference image containing vegetation elements (top-left corner) and parts from other rocks (bottom-right corner)	53
5.4 Comparison between an original reference image and its version without the unwanted background elements	53
(a) Original reference image	53
(b) Reference image with unwanted elements removed from the background	53
5.5 Comparison between the feature set extracted from an original reference image and its version without the unwanted background elements	54
(a) Feature points extracted from the original reference image	54
(b) Feature points extracted from the reference image without the unwanted background elements	54
5.6 Diagram that shows the flow of preprocessing the camera input before using it for tracking	56
5.7 Outline extracted from the engraving representing a horse	58
5.8 Features extracted from the horse engraving outline	59
5.9 Features extracted from one of the goat engraving outlines	59
6.1 Question QA1 Results	70
6.2 Question QA2 Results	71

6.3	Question QA3 Results	71
6.4	Question QA4 Results	72
6.5	Question QA5 Results	72
6.6	Question QA6 Results	73
6.7	Question QB1 Results	76
6.8	Question QB2 Results	76
6.9	Question QB3 Results	77
6.10	Question QB4 Results	77
6.11	Question QB5 Results	78
6.12	Question QB6 Results	78
6.13	Question QB7 Results	79
6.14	Question QC1 Results	81
6.15	Question QC2 Results	81
6.16	Question QC3 Results	82
6.17	Question QC4 Results	82
6.18	Question QC5 Results	83
6.19	Different scales for interpreting a SUS score,	84

List of Tables

2.1	Results of the study by Swan et al. [13]	8
2.2	Comparison of AR SDKs	11
3.1	Non-Functional Application Requirements	23
3.2	Functional Application Requirements	23
3.3	Tracking states on each application page	24
4.1	Features comparison between Vuforia and ARFoundation	33
4.2	Number of reference images needed by the different image detection approaches tested, with n being the number of images needed for the detection of a single object and x the number of engravings in the target rock	41
4.3	Texture offsets used to display each frame of an animations containing 3 frames	46
5.1	Quality scores obtained by each artificially generated reference image, calculated with the <i>arcoring</i> tool	50
5.2	Performance metrics results for the approach of using multiple reference images for each engraving/rock, including 3 different angles and 5 different pictures for each angle	51
5.3	Number of frames processed per second by the application using each approach	51
5.4	Performance metrics results for the approach of using multiple reference images for each engraving/rock, including 2 different angles and 3 different pictures for each angle	52
5.5	Quality scores obtained by each version of the reference image	54
5.6	Performance metrics results for the approach of using multiple reference images for each engraving/rock	54
5.7	Number of frames processed per second by the application using each tested approach	55
5.8	Average frames per second achieved by the application using each approach, measured on a Xiaomi Redmi 10T Lite phone	56
5.9	Performance metrics results for the approach of increasing the contrast of the frames captured by the camera	57
5.10	Performance metrics results for the approach of increasing the brightness of the frames captured by the camera	57
5.11	Quality score obtained by each outline, calculated with the <i>arcoring</i> tool	60
5.12	Number of matches achieved by using each outline as reference image	60
5.13	Performance metrics results for the approach of increasing the brightness of the frames captured by the camera	61
5.14	Performance metrics results for the approach of using an external flashlight to increase the luminance of the environment	61

6.1	Task A steps and questions	65
6.2	Task B steps and questions	66
6.3	Number of participants of each gender	67
6.4	Number of participants of each age range	67
6.5	Statistics regarding participants' age	67
6.6	Number of participants of each age range	67

Abbreviations and Symbols

AR	Augmented Reality
VR	Virtual Reality
HMD	Head-Mounted Display
MAR	Mobile Augmented Reality
PAR	Pervasive Augmented Reality
SDK	Software Development Kit
NFT	Natural Feature Tracking
GPS	Global Positioning System
UI	User Interface
SUS	System Usability Scale

Chapter 1

Introduction

Museum and cultural heritage site visits are often one-time experiences targeted at specific audiences, which limits the reach of some exhibitions and diminishes the interactivity with the general public. One way to improve on this matter is by introducing Augmented Reality (AR) into museum and cultural heritage site visits, opening up a new range of possibilities to use the same physical content to provide different experiences to visitors, by allowing users "to see the real world, with virtual objects superimposed upon or composited with the real world" [3]. The aim of this dissertation is to develop a robust mobile application that, by using AR technologies, can help improve the experience of the Vale do Côa rock engravings visitors. The improvements might come in the form of providing users with an easy way to identify and visualize complex engravings and providing extra information about them digitally in the form of text or animations that can provide users with a story and a context for each specific engraving. Since these visits happen in an outdoor setting, this dissertation also aims to contribute to the exploration of new outdoor calibration settings for AR applications.

1.1 Context and Motivation

This dissertation comes from a partnership between InfoPortugal¹, a digital content and technological geolocation solutions company that focuses mostly in the area of tourism, the Côa Museum², which aims to preserve the prehistorical art of the Côa valley, and the Faculdade de Engenharia da Universidade do Porto, the institution where this dissertation is developed.

In a world where culture is of major importance to society, having the ability to provide custom experiences to museum visitors becomes a critical step in the dissemination of cultural knowledge across multiple areas. In the Côa Valley, rock engravings are located outdoors scattered across different locations with various accessibility levels and may appear overlapped with each other, becoming harder to identify and visualize by visitors on their own. Furthermore, visiting the engravings is currently a one-time experience, meaning that if someone wants to revisit them,

¹InfoPortugal homepage: <https://infoportugal.pt/>

²Côa Museum homepage: <https://arte-coa.pt/>

the content will be the same as on the last visit. To address these problems, mobile AR can be introduced to museum visits. In the case of the Vale do Côa museum, an AR app would allow visitors to better identify engravings since they could be highlighted on the app when pointing the camera at the rock they are inserted in, to receive extra information on the physical content via text, video or images, and to customize their experience by choosing which content to receive on the app. Moreover, it would be beneficial for the museum itself as the content of visits could be renovated and changed periodically through the use of animations and other digital content. On the technical side, this dissertation aims to explore the issues and limitations of applying Augmented Reality to flat rock engravings.

1.2 Research Questions

This dissertation aims to answer the following research questions:

- **RQ1: What are the issues of applying AR to flat rock engravings, especially in outdoor conditions, and how can these issues be mitigated?**
- **RQ2: Can a mobile AR application help rock engravings visitors better visualize and understand rock art in an interactive, intuitive and simple way?**
- **RQ3: Can AR improve the experience of the Côa Valley engravings visitors and is it feasible to apply AR to that context?**

1.3 Objectives

The objectives of this dissertation are the following:

- Identify the limitations and issues of the application of Augmented Reality to flat rock engravings.
- Prototype and test a mobile Augmented Reality application that explores AR digital content to be applied on top of rock engravings.
- Evaluate how a mobile Augmented Reality application can improve the cultural experience of the Côa Museum visitors

1.4 Document Structure

Chapter 1 contains an introduction to the dissertation, explaining its context and motivation, the research questions it tries to answer and the objectives of the project. This is followed by chapter 2 which contains the bibliographic review, exploring AR in general as well as its application to tourism and culture. In addition, work related to this dissertation is researched and presented in this chapter. The developed solution's architecture and specifications are detailed in Chapter

3, including the application requirements and the user interface design. Chapter 4 describes the solution, providing implementation details and approaches used to improve the application's performance. The solution was evaluated and the results are presented in chapter 5, along with an analysis. Finally, chapter 7 states conclusions regarding the work carried out for this dissertation, as well as improvements and future steps in the development of the solution.

Chapter 2

Bibliographic Review

This chapter goes through the available information and state-of-the-art related to the scope of this dissertation.

2.1 Introduction

Although Augmented Reality is widely researched, its application to outdoor scenarios over rock surfaces, often not entirely flat, is not a thoroughly investigated area and presents growth and development potential.

The first section of this chapter will focus on AR as a technology, defining it and detailing its types and areas of application. Furthermore, it will cover the currently available AR technologies and frameworks and highlight future developments and investigation ideas for this field of research. The application of AR to the tourism and culture fields will be explored in the second section, focusing on the improvements AR applications can bring to cultural experiences, especially rock engraving visits. Finally, the third section will discuss the current state of outdoor scenarios application of AR.

2.2 Augmented Reality

2.2.1 Definition and History

The term "Augmented Reality" was first introduced to the world in the 90s with the work of Tom Caudell [1] for Boeing Computer Services ¹, introducing the idea of having digitally assembled information being presented to the user as part of the real world. In this case, the goal was to enable "a factory worker to view index markings or instructions as if they were painted on the surface of a workpiece" [1], using a Head-Mounted Display to track the user's head position and orientation. One of the main advantages of AR, when compared to Virtual Reality (VR), presented in Caudell's work [1], was that it is much less expensive computationally since there is no need

¹ Boeing Homepage: <https://www.boeing.com/>

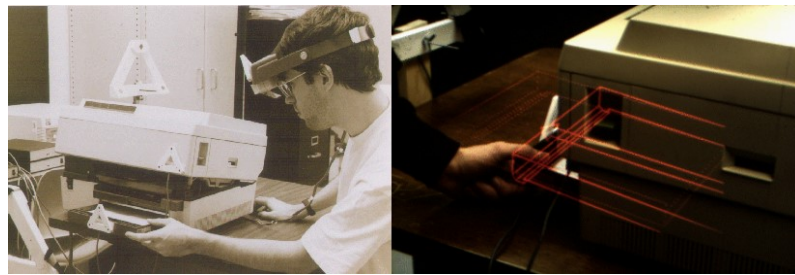


Figure 2.1: The KARMA prototype

to render a complete virtual environment, but only parts of it. The authors noted hardware design, difficulties in position and orientation sensing, and, most importantly, registering a user and its sensors' positions as significant issues in developing complex AR systems that required further research.

In 1993, Steve Feiner [15] developed the KARMA (Knowledge-based Augmented Reality for Maintenance Assistance) prototype system which proved to be a major contribution to the field of AR since it was the first practical system that demonstrated the potential benefits of AR applications. This system relied on a see-through HMD with a small display capable of generating low-resolution red virtual images and a set of three-dimensional trackers to provide the user with virtually generated instructions to perform maintenance actions on a laser printer. These instructions would be shown in the HMD's display, overlapping the user's vision and appearing as if placed in the real world. Figure 2.1 shows the HMD used in the KARMA project as well as the graphics presented to the user virtually as instructions.

The most consensual definition of AR arose in 1997 by Ronald Azuma [3], who broadly defined AR as "a variation of virtual environments (VE), or virtual reality as it is more commonly called". According to Ronald Azuma [3], an AR system has three characteristics:

1. Combines real and virtual environments.
2. Incorporates real-time interaction.
3. Is registered in the three-dimensional space.

In 1994, AR was used in the entertainment industry for the first time by producer Julie Martin who created a theatrical performance called "Dancing in Cyberspace" in which dancers would interact with virtual objects on stage and it was in 1998 that AR was first introduced to the sports field when the NFL placed a virtual yellow line in a broadcast denoting the first down line.

The first AR framework, called ARToolKit² was developed by Kato and Billinghurst in 1999 [24] and released to the public as an open-source project in 2001.

In a paper published in 2000, Thomas et al. [40] present the first application of AR to the gaming industry with the ARQuake³ project. This project aimed to produce a version of the

²ARToolKit Homepage: <http://www.hitl.washington.edu/artoolkit.html>

³ARQuake Homepage: <http://www.tinmith.net/arquake/>

popular desktop game *Quake* based on AR where the user would use an HMD to see the game's objects, like monsters, in the real world, as shown in figure 2.2. ARQuake used ARToolKit [24] as the basis of its optical tracking system which relied on fiducial markers, "known-sized squares with high contrast patterns in their centres" [40] to calculate the orientation and distance of objects. Although the project was not completed at the time of paper submission, ARQuake continued to be developed and was eventually released to the public.



Figure 2.2: ARQuake game

The year 2006 saw the release of the first mobile AR game, ARTennis, developed by Henryson et al. [20], available only for Nokia phones. The game used a set of markers on a table to determine each player's phone position and orientation relative to the table and to display a virtual tennis court on the players' phones, which can be observed in figure 2.3. The user feedback received by this project was very positive, which contributed to the subsequent rise of mobile AR's popularity.

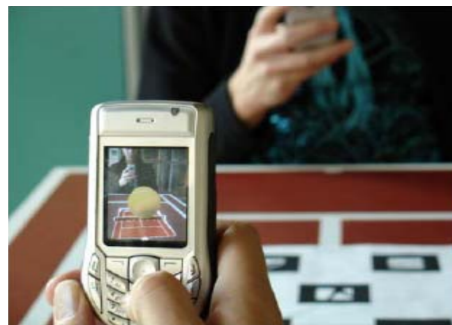


Figure 2.3: ARTennis application on a Nokia device

From then on, the area of AR continued to grow with new features being introduced and device support being increased, reaching a point where almost any mobile device available in the market today is capable of running AR applications. Although the central area of growth was mobile AR, pervasive AR, which usually relies on wearable devices like Head-Mounted Displays, has also been growing with improvements being applied to these devices. One such example is the work carried out at Cambridge University by Shrestha et al. [38] that focused on improving the comfort of HMDs, which led to a significant reduction in the side effects of using these devices.

2.2.2 Types of AR

AR applications can be classified into different types according to the type of display, the way the camera's pose is estimated, and the kind of tracking used.

2.2.2.1 Display and Tracking

According to the type of display device used by an AR application and the type of tracking used, it can be classified into two types, Mobile Augmented Reality (MAR) [22] or Pervasive Augmented Reality (PAR) [19].

MAR is based on mobile devices, like tablets or smartphones, using their cameras and sensors to capture and shape the real-world environment. In recent years, hardware improvements to mobile devices have significantly contributed to the growth of MAR, enabling mobile devices to perform essential tasks to AR applications, like tracking and registration operations [34]. The goal of MAR applications is to provide the user with short, interactive and mobile AR experiences without always considering the context of interactions.

On the other hand, PAR aims to provide users with a "continuous, omnipresent, and universal augmented interface to information in the physical world" [19]. For this, PAR systems need to capture the context where the user is inserted and adapt to it. Changes to the user's context should also introduce changes to the AR system's behavior, to make the experience as smooth and seamless as possible. PAR makes use of Head-Mounted Displays, sensors and wearable computers [40] as its hardware basis to achieve higher accuracy and performance than MAR technologies, although losing on the mobility aspect. Furthermore, using HMDs improves the user's comfort in long AR experiences.

2.2.2.2 Pose Estimation

In the work developed by Zendjebil I. et al. [46] in 2008, AR is classified into two types based on the camera pose estimation technique: Marker-based and markerless methods.

Marker-based methods use markers to estimate the position and orientation of the user's camera. The markers have to be easily detected and the application needs to know their positions a priori to be able to estimate the camera's pose. Markers are usually square or circular and include a pattern with a code used for recognition. Figure 2.4 shows circular and square markers examples. Gherghina et al. [18] propose a mobile AR system that uses QR Codes as markers, projecting digital content on top of those markers. Marker-based methods became popular with the ARToolKit framework, achieving great success in the AR community. Zendjebil et al. [46] point to the need for artificial markers to be placed in the real world as the main disadvantage of marker-based methods since it might not be possible to do so.

On the contrary, markerless methods rely on natural features already present in the real-world environment, do not require any artificial markers and detect a real-world object by their geometrical parameters, such as corners or edges. The camera's pose is then estimated concerning the natural features seen in the real-world environment. Huang et al. [21] used markerless AR



Figure 2.4: Examples of markers used in marker-based methods

techniques to develop a piano teaching system in which the main features of the piano were detected and used to estimate the user's position relative to the piano itself. The main advantage of markerless methods is that there is no need for artificial markers, widening the range of application for markerless AR applications. However, as pointed out by Chen et al. [9], these methods require higher computational complexity, which may limit the range of devices capable of using markerless AR techniques.

2.2.3 Areas of Application

Over the years, Augmented Reality has proved to be a powerful technology being applied in many different areas, improving user experience in various ways. According to the study carried out by Swan et al. [13], in which the authors analyzed papers related to AR that involved user studies, there are nine main application areas for AR: 1) collaboration, 2) education, 3) entertainment and gaming, 4) industrial, 5) interaction, 6) medicine, 7) navigation and driving, 8) perception, and 9) tourism and exploration. Table 2.1 shows the number of papers reviewed by the authors as well as the most common type of display used in each AR application area.

Area of Application	Number of papers	Most common type of display
Collaboration	12	HMD/Handheld/Desktop
Education	42	Handheld
Entertainment and Gaming	14	Handheld
Industrial	30	HMD
Interaction	67	HMD/Handheld
Medicine	43	Desktop
Navigation and Driving	24	Handheld
Perception	51	HMD
Tourism and Exploration	8	HMD/Handheld/Desktop

Table 2.1: Results of the study by Swan et al. [13]

Out of these areas, perception and interaction are the broadest, including AR projects in a range of different domains and with various goals. Along with these two, Education and Medicine are the most researched areas. On the other hand, Tourism and Exploration and Collaboration are the less investigated and researched categories. According to the authors, these areas present many

opportunities for user studies and AR projects. Handheld devices are the most common type of display in the majority of areas. HMDs are the most used display devices in the Industrial area, due to most applications in this domain requiring the use of both hands, making handheld devices unusable. In the Perception domain, HMDs are preferable when the goal is to offer users a highly immersive experience, as is the case in most projects in this area. In the Medicine area, desktop displays are the most common, not disturbing the doctor's view and not adding any additional load. This dissertation is included in the Tourism and Exploration area, one of the less researched categories. In this area, most projects were museum-related and were usually carried out in indoor scenarios. The authors note the potential evolution of this area for AR applications, with handheld devices being the preferred type of display due to them being currently easily accessible and easy to use.

2.2.4 Augmented Reality Systems

Augmented Reality systems contain four key components, as mentioned by Rodrigues [36]: 1) camera, 2) display device, 3) graphics system, and 4) tracking system. The camera is responsible for capturing the real-world environment and can be integrated into the display device, like smartphones, tablets, or external devices. The display device is used to show the user the AR experience's result, combining the real-world information captured by the camera and the digital content displayed on top of it. The graphics system computes all the generated graphics and uses the information provided by the tracking system, which tracks the user's position and orientation in relation to the scene, to render virtual objects correctly.

2.2.5 AR Hardware

According to the study developed by Kim et al. [26], AR hardware platforms can be classified into five different categories: 1) computers with webcam, 2) smartphones, 3) Head-Mounted Displays, 4) glasses, 5) haptic devices.

The combination of computers and webcams is the most accessible AR setup for most people. However, it offers little mobility to the AR experience due to the fixed nature of computers. Because of this, AR applications that use computers and webcams typically use marker-based methods for pose estimation. Mobile devices, including smartphones and tablets, are currently the most common way to access AR experiences using marker-based and markerless methods. Despite these devices offering excellent mobility to the AR application, they may lack computational resources compared to other hardware platforms. However, mobile devices are becoming a more prominent source of computational power, and the tendency is for the mobile industry to continue rising and for devices to improve. HMDs are used in long and continuous AR experiences due to the better comfort they can provide to users during more extended periods of use compared to, for example, mobile devices. Since they extend the user's vision, HMDs are great for immersive experiences. The most significant disadvantage of these devices is their need for external equipment

and sensors, which can limit the AR experience to small areas. Smart glasses are still an uncommon technology, although they have been gaining market traction. This type of device aims to give the user a continuous and personalized experience based on their context. Finally, haptic devices are a standard technology in video game consoles like the Nintendo Wii or the PlayStation4 and extend the AR experience with synthetic haptic interactions, like tactile feedback, enabling the user to touch virtual objects.

2.2.6 AR Software

AR Software Development Kits (SDKs) were created to facilitate AR applications' development and allow developers to abstract their work from the low-level aspects of the technology. An analysis of some of the AR SDKs available in the market will be presented in this subsection, comparing various aspects of the most common ones. Since this dissertation aims to develop a mobile AR application, the analysis will focus only on SDKs that support mobile development and are compatible with the Unity⁴ game engine since it will be used to develop the mobile application. The criteria used for the comparison were the following:

- Price
- Documentation
- Supported Tracking Methods
- Compatibility with Unity
- Supported Platforms

Table 2.2 compares seven AR SDKs (ARToolKit⁵, Vuforia⁶, Wikitude⁷, Kudan⁸, ARCore⁹, ARKit¹⁰ and EasyAR¹¹) based on these aspects.

ARToolKit, Vuforia, Wikitude, and ARCore are the most robust and efficient SDKs for this project and all have been used successfully in the past in similar projects [8, 37, 17]. Wikitude is the most complete SDK explored, including various advanced features like location-based tracking using the phone's GPS sensor and compass, but its main disadvantage is that the free version includes a watermark in the camera view, needing a paid license for it to be removed and for the app to be deployed. On the other hand, ARCore is completely free, has extensive documentation supported by a big community background, and is targeted at beginners. Both ARToolKit and Vuforia display great performance when applied to rock engravings, as shown by Blanco-Pons et al. [7], and are capable of handling outdoor conditions well.

⁴Unity Homepage: <https://unity.com/>

⁵ARToolKit Homepage: <http://www.hitl.washington.edu/artoolkit.html>

⁶Vuforia Homepage: <https://developer.vuforia.com/>

⁷Wikitude Homepage: <https://www.wikitude.com/>

⁸Kudan Homepage: <https://www.kudan.io/>

⁹ARCore Homepage: <https://developers.google.com/ar?hl=en>

¹⁰ARKit Homepage: <https://developer.apple.com/augmented-reality/arkit/>

¹¹EasyAR Homepage: <https://www.easyar.com/>

SDK	Price	Documentation	Tracking Methods	Unity	Platforms
ARToolKit	Free	Extensive	Marker-based Markerless	Yes	Android iOS
Vuforia	Free/Paid Plans	Extensive	Marker-based Markerless	Yes	Android iOS
Wikitude	Free/Paid Plans	Extensive	Marker-based Markerless	Yes	Android iOS
Kudan	Free/Paid Plans	Limited	Marker-based Markerless	Yes	Android iOS
ArCore	Free	Extensive	Marker-based Markerless	Yes	Android iOS
ARKit	Free	Extensive	Marker-based Markerless	Yes	iOS
EasyAR	Free/Paid Plans	Limited	Marker-based Markerless	Yes	Android iOS

Table 2.2: Comparison of AR SDKs

In order to decide which SDK to use in this project, these four SDKs will have to be tested and compared thoroughly in the context of the Vale do Côa engravings.

2.3 AR Applied to Tourism

This chapter starts by discussing the use of mobile AR applications in the tourism domain, followed by an analysis of the benefits of MAR in cultural visits.

As presented in section 2.2.3, Tourism and Exploration is one of the areas of application of AR. Although it is a less explored and researched area than others, like Education or Gaming and Entertainment, this application area has been growing in interest recently as the importance of digital innovation in tourism is becoming more noticeable. The augmented reality experience in tourism is defined by Yovcheva et al. [45] as a "complex construct which involves the emotions, feelings, knowledge and skills resulting from the perception, processing and interaction with virtual information that is merged with the real physical world surrounding the tourist" and Garcia-Crespo et al. [16] state the need for dynamic and integrated services in the tourism industry.

Tomicic et al. [25] identify five dimensions that directly influence the success of AR applications in the tourism domain: 1) general requirements, 2) functionalities, 3) issues, 4) overlay types, and 5) technologies. The critical requirements for tourism AR apps include the application's dynamics, interactivity and entertainability, the connection between the user's context and the content provided by the application, and an intuitive user interface with simple navigation. The functionalities should include browsing and searching for specific content, map browsing, routing and navigation, and a direct connection to service providers and other users. The overlay types refer to the media to be digitally rendered on top of the real-world environment and include

text, graphics, videos, sounds and haptic interactions. The issues dimension is related to the hardware requirements, user safety and comfort, and accessibility. Finally, technologies refer to the characteristics of the devices used by the applications, like the available sensors and the size and resolution of the screen for mobile devices.

Williams et al. [44] propose and evaluate a MAR application, ToARist, capable of showing the user tourism routes and points of interest, augmenting them with digital information, as shown in figure 2.5. The results obtained were positive, with several users claiming their increased interest in exploring certain areas if they had access to the application.

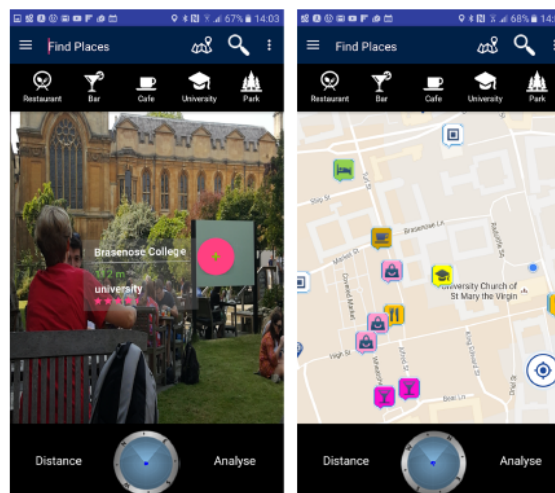


Figure 2.5: ToARist MAR application

2.3.1 Mobile AR Applied to Cultural Visits

Cultural sites represent valuable sources of knowledge and history that can highly influence the visitor's perception of the world and society. Furthermore, De Benedictis et al. [12] mention the importance of cultural experiences in the development of creativity and in enriching knowledge, denoting cultural experiences as "a stimulating starting point for learning". Simple physical visits have been decreasing in popularity, especially among the younger generations, which led cultural organizations to look for alternative ways to improve the visitor's experience. Both VR and AR arise as possible solutions to this problem, increasing the visitor's immersion and interactivity with the cultural site and, ultimately, improving the overall experience. Although VR can achieve higher levels of immersion than AR, it is also seen as a threat to travel, as stated by Cheong [10], since one can recreate any tourist destination using VR technologies. For this reason, AR seems more suitable for cultural visits, enabling the visitor to interact with the natural environment while still being able to visualize digital content. As mentioned by Liang & Elliot [30], "AR might be more suited to enhancing (co-creation) rather than replacing the visitors' experience" and enables users to experience cultural items in different ways [6].

According to Vlachos et al. [42], AR applications in the cultural heritage domain can be classified into five types: 1) educational, providing historical knowledge about the cultural site to the

user, 2) exhibition, to increase the immersion and overall experience of content visualization, 3) exploration, allowing the user to explore current and older versions of the site, 4) reconstruction, providing interaction with historical versions of the cultural site by offering digital reconstructions of it, and 5) virtual museums, presenting information about the cultural site in a museum environment.

The application of AR to cultural visits has been explored in different contexts. Falomo et al. [14] developed a mobile AR application for the Pavia University History Museum¹² to provide additional information regarding the exhibited items to visitors. Users could download the application to their smartphones and use the markers placed at multiple points of the museum to trigger the virtual display of information about specific museum items on their phones. The results were positive as visitors showed appreciation for the app and primarily provided positive feedback regarding the application's usability and usefulness. Figure 2.6 demonstrates how the MAR application works.

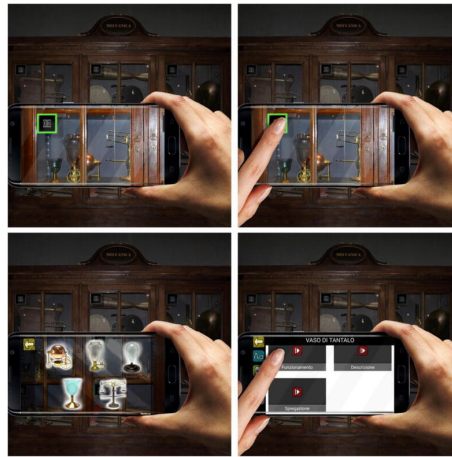


Figure 2.6: AR application for the Pavia University History Museum

Directed more towards open cultural heritage sites rather than indoor museums, the ARCHEOGUIDE project [43], by Vlahakis et al., presents a mobile AR application that aims at personalizing cultural site visitors' experiences with a virtual guide and tour assistant. The authors tested the application in the ancient city of Olympia, Greece. Apart from providing on-site help to visitors, the application can display reconstructions of ruins and monuments as well as animations. One example is the display of collective sports animations at the stadium of Olympia, which shows the users how various sports were played in ancient times using digital representations of athletes, as shown in figure 2.7.

2.4 Related Work

The application of AR to archaeology is a fast-growing area that has seen a significant increase in research volume over the past decade. In rock art sites, AR has been used to explain and annotate

¹²Pavia University Museum Homepage: <http://musei.unipv.eu/msu/home-eng/>



Figure 2.7: Virtual athletes running in the Olympia stadium

rock engravings, thus enhancing the visitors' experience.

There are three main issues related to AR applied to rock engravings: 1) which type of pose estimation method should be used: marker-based or markerless, 2) overcoming unstable and changing outdoor conditions, and 3) detecting rock engravings at different distances and angles. The following subsections will further explore these three aspects in the following subsections.

2.4.1 Pose Estimation: Marker-based or Markerless

Rock art sites are usually protected places with restricted access for preservation purposes. This invalidates marker-based methods as an option for pose estimation since placing artificial markers would alter the site's environment. Therefore, markerless methods are preferable for this type of application.

Markerless methods can be divided into two categories [7]: model-based and feature-based. While model-based methods use 3D models of the scene to estimate the user's pose and are usually found in the context of museums or controlled environments [11, 31], feature-based methods [8, 33], also called Natural Feature Tracking (NFT), rely on the detection and recognition of natural features present in the real environment. Since rock art is usually found in flat rocks, NFT is the preferred method for AR applications in this area since the stones can generally be assumed as 2D planes in a 3D environment. The challenge with markerless methods applied to rock art is, according to Blanco-Pons et al. [7], "to recognise enough features in a real environment to determine the pose and orientation of the camera in real time", which becomes complex in most rock art sites due to the faintness of engravings and poor lighting conditions often found in these places.

2.4.1.1 Natural Feature Tracking

Natural Feature Tracking (NFT) compares and matches the features gathered in real time by the user's device to previously computed feature vectors obtained from reference images. Therefore, NFT requires a dataset of reference images from which keypoints are extracted to be stored offline

in a feature database. NFT methods are divided into three phases [41]: 1) extraction, 2) description, and 3) matching.

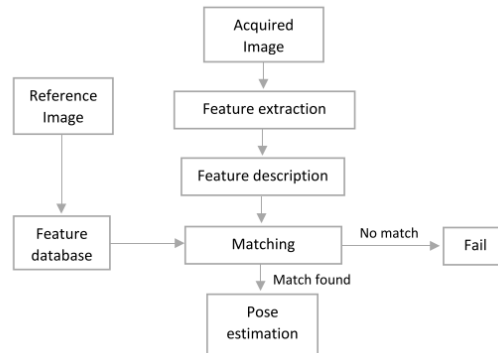


Figure 2.8: NFT system

As shown in figure 2.8, there are two different pipelines until the matching phase, one for the reference image and the other for the real-time acquired snapshot. The reference image is obtained offline and used to extract the natural features of the scene, which are then stored in a feature database as vectors. The real-time acquired snapshot starts by going through a feature extraction or keypoint detection phase, with the goal of "finding pixels which have different appearance from other pixels", as mentioned by Uchiyama and Marchand [41]. Following that, in the feature description phase, a feature vector is computed from the keypoints gathered previously. Finally, both pipelines join in the matching phase, in which the keypoints obtained from the acquired image are compared and matched to the features extracted from the reference image. If the feature database does not contain a feature vector similar enough to the keypoints found in the acquired image, the matching phase fails. On the other hand, if a match is found, the user's pose can be estimated.

2.4.2 Handling Outdoors Conditions

Rock art can be found indoors, usually in museum settings, and outdoors, in their original location. While AR applications easily handle indoor scenarios due to the controlled nature of the environment and its conditions, using AR outdoors is a significantly more difficult and complex task. Ronald Azuma [4], in 1999, determined three major obstacles to the application of AR in outdoor scenarios: 1) size, weight and power issues with the devices used, 2) display settings, and 3) tracking. These three issues will be explored in detail in the following subsections.

2.4.2.1 Size, Weight and Power Issues with AR Devices

The use of AR in outdoor scenarios usually requires users to move around, requiring that the devices used for the AR experience are easily transportable and comfortable to use. This used to be a problem when AR was mostly based on computers and sensors due to the lack of mobility offered by these devices. However, nowadays mobile devices have enough computational power and are

equipped to execute AR applications easily, presenting a lighter, smaller and mobile alternative to computers and external sensors. Although mobile devices are not able to reach the levels of tracking accuracy of fixed AR setups, most outdoor AR applications can support some tracking inaccuracies, making mobile devices like smartphones and tablets the preferred type of device for outdoor AR experiences.

2.4.2.2 Display Settings

In outdoor environments, conditions cannot be controlled, as mentioned by Blanco-Pons et al. [7], which affects the tracking of features and increases the load on the AR display, since it needs to be able to work across a range of different conditions. Since engravings are situated on rock surfaces, their surroundings usually present similar colors, increasing the difficulty of their detection and visualization, which becomes an even more complex task once the scene's brightness and contrast are taken into account. One way to mitigate the unstable nature of these conditions is, for each rock engraving, to populate the reference images dataset with pictures taken in different states. For example, in the approach used by Tsigkas et al. [33], pictures were collected at four different times of the day with varying lighting settings. From a different perspective, Blanco-Pons et al. [7] artificially implemented the same idea by editing the pictures in Adobe Photoshop¹³ and changing their exposure, therefore simulating different lighting conditions.

Both studies mentioned before [7, 33] achieved outstanding results in handling outdoor conditions. While Blanco-Pons et al. were able to mitigate the problems introduced by varying lighting settings and occlusion of the target, Tsigkas et al. explored other types of outdoor conditions apart from lighting, such as wet weather and the presence of falling leaves. From these two studies, we can conclude that the existing AR frameworks are capable of handling outdoor conditions, with good prospects regarding the evolution of technology in this area.

2.4.2.3 Tracking

The tracking of objects in a scene and the user's position can be achieved with great precision in indoor settings due to the controlled nature of the environment. The objects to be tracked can be modified to be more easily distinguishable from the background or the background itself can be changed to offer better tracking conditions. However, this is often not the case in outdoor scenarios, as the background can usually not be changed and there might be restrictions to the modifications that can be done to objects. One example of this, mentioned by Ronald Azuma [4], is a military AR application, in which would be unrealistic to have soldiers wear bright clothes or stamp markers in their uniforms to help the AR tracking system. However, there are some ways to improve tracking in outdoor scenarios.

- Vision-based tracking: Based on object recognition techniques, vision-based tracking makes use of 2D images to detect and recognize objects in the scene. This technique extracts features from the provided images which are then matched to 2D views of 3D models, as

¹³Adobe Photoshop Homepage: <https://www.adobe.com/pt/products/photoshop.html>

mentioned by Beier et al. Through the matching of the extracted features with the camera view, captured in real time, it is possible to determine the camera's trajectory in relation to the scene [39]. The main disadvantage of this tracking method is its inherent slower performance when compared to other tracking techniques, like inertial tracking. Although this is a critical problem for applications that require the user to perform fast movements, it does not pose problems to applications that only require limited movement of the camera.

- **GPS and compass:** One way to determine the user's position and orientation is to use the GPS sensor present in most mobile devices nowadays along with its compass. GPS can be used to determine the user's coordinates in the world while the compass can calculate the device's orientation. This approach was used by Zhang et al. [47] to track rock paintings located in high positions in the Helan mountain, China. Although 67% of the users that tested the application developed by Zhang et al. felt that the application was stable and presented high accuracy, the authors denote the lack of precision achieved by the GPS technology. Furthermore, since compasses work by measuring the Earth's magnetic field, they are prone to error in the presence of other magnetic fields, providing incorrect orientation results, which might pose a problem in some locations.
- **Inertial tracking:** Based on accelerometers and gyroscopes, inertial tracking is mostly immune to environmental changes. Accelerometers obtain the user's three translational coordinates while gyroscopes provide the rotational angles, achieving a tracking system with six degrees of freedom. Inertial tracking has the advantage of providing great results when fast movements occur, contrary to, for example, vision-based tracking, which is slower. The problem with inertial sensors is the accumulation of error with time, as mentioned by Azuma [4]. Sensor noise and drift can lead to small orientation and position errors that might not pose problems to the tracking system in an initial stage, but after some time they may accumulate, increasing the overall error of the system by a significant amount. Lang et al. [27] propose an inertial tracking system for MAR. In the article, the authors discuss a way to correct the drift generated by the inertial sensors using an external vision-based tracking system to compensate for the inertial error.
- **LiDAR sensors:** According to Li et al. [28], "Light detection and ranging (LiDAR) technology enables the accurate determination of an object's distance (and velocity) information". Although this technology is widely used in areas like robotics and meteorology, its application to AR is still rather unexplored due to the unavailability of integrated LiDAR sensors in devices used for AR applications, like smartphones or computers. LiDAR technology is able to construct a point clouds of the environment, which can later be used to match with features extracted from 2D images, thus tracking the device's camera position [29]. The main problem with this technology is, as mentioned before, the unavailability of integrated sensors in most devices, leading to the need of external sensors in order to use LiDAR for

AR. However, such integrated sensors are already available in the most recent Apple smartphones and tablets and are currently used to enhance these devices' AR capabilities with great success.

- **Hybrid Techniques:** Hybrid tracking techniques try to join the advantages of vision-based and inertial tracking while mitigating the issues of both methods. As mentioned by Ribo et al. [35], "Vision-based tracking systems often fail to follow fast motion, especially as the camera rotates", even though they are very stable and accurate on the long term. On the other hand, inertial tracking systems can handle fast camera motions but lack stability over longer periods of time due to sensor noise and drift. The idea of hybrid tracking is to fuse the two methods to compensate for each of their weaknesses. The vision-based system can be used to correct the inertial tracking's sensor drift while the inertial system is used to aid the vision-based method with fast camera movements [23].

2.4.3 Detecting Rock Engravings at Different Distances and Angles

Rock engravings are found in various locations in the wild, some of them inaccessible to visitors, which introduces a critical problem to their accurate detection: the AR application needs to be able to detect engravings at different distances. Most engravings are found near the ground, where visitors can get close to and use the AR application at short distances. However, some engravings are located in remote places, whether in very high spots or areas of dangerous access to visitors. Zhang et al. [47] propose a solution for this problem that uses the GPS and compass sensors available in smartphones to calculate the device's orientation for long-distance engravings. The application developed by Zhang et al. for this study [47] is divided into two stages. The first stage consists of a navigation map that shows the user where each rock painting is located. In the second stage, paintings are divided into two types based on their distance to the user, and a distance threshold is used to classify paintings as short-distance or long-distance. NFT is used to detect and recognize short-distance paintings since this method presents great accuracy in short distances. On the other hand, for paintings located at longer distances, it is impossible to capture them with enough resolution for the NFT method to work correctly. To solve this issue, the pitch and yaw of the user's smartphone are obtained from its compass and turned into a direction vector. This is then used to calculate the angle between the mobile device and the painting, determining whether the user is oriented toward the target painting. In a study carried out in 2019, Blanco-Pons et al. [7] investigate and compare the performance of two AR frameworks, Vuforia¹⁴ and ARToolKit¹⁵, at detecting images over multiple distances, concluding that the Vuforia SDK is faster and more stable across all the distances tested.

¹⁴Vuforia Homepage: <https://developer.vuforia.com/>

¹⁵ARToolKit Homepage: <http://www.hitl.washington.edu/artoolkit.html>

2.4.4 Examples

Many different studies have been conducted in this area [32, 8, 2, 33, 37], considering different approaches and applications of AR to rock art sites across the world. Mosha et al. [32] propose a MAR application for the rock art located in the Brandberg mountain, in Namibia, aiming to complement the local tour guides, give users more context on the origin and history of the rock art pieces and provide visitors with virtual reconstructions of rock engravings that are currently damaged or fading away. Blanco-Pons et al. [8] designed and implemented an application for the Cova dels Cavalls rock art in Valencia, Spain, achieving excellent results among users, who, in their majority, claimed that the application helped better visualize rock paintings and understand the context of the scene. Abergel et al. [2] explored the use of AR for rock art annotation using the SLAM algorithm, testing and evaluating the developed solution on around 180 French rock art sites. Tsigkas et al. [33] focused their work on the archaeological site at Vathy, Astypalaia, Greece, exploring ways to detect rock engravings using markerless AR methods. Two detection methods were tested and compared, one based on a deep learning approach and the other based on feature points detection. Zhang et al. propose "an outdoor augmented reality guidance system for ancient rock painting aiming to enhance tourist experience and introduce the culture and historical background" [47]. The developed mobile AR application targets the rock paintings in Helan Mountain, China, and includes a navigation map showing the visitor where paintings are located. Furthermore, it accurately identifies rock paintings in the wild and overlaps digital content, like annotations or images, on top of them in an interactive way. Finally, Aliisa Råmark [37] investigated how AR could be applied to the Tanum World Heritage¹⁶ site, in Sweden, as a means to increase accessibility and heritage value.

2.5 Conclusions

Tourism and culture can benefit significantly from AR applications, increasing the interactivity of cultural visits across all age groups. Although there are some studies about the application of AR to these domains, AR research has primarily been focused on other areas, like education or entertainment. Since rock engravings are not always clear to visitors, rock art sites are great candidates for MAR applications that can improve the visualization and understanding of such art.

As mentioned in this document, the application of AR to rock engravings has three main problems: pose estimation methods, handling outdoor conditions, and detecting rock engravings at different distances and angles. To decide which pose estimation method to use, on-site tests will have to be carried out to understand the feasibility of using markers and compare the performance of marker-based and markerless methods in different conditions. Regarding the handling of outdoor conditions, reference images will be collected on-site at different times of the day and with varying settings of lighting to improve the accuracy of feature detection and different tracking

¹⁶Tanum World Heritage site Homepage: <https://whc.unesco.org/en/list/557/>

methods will be tested. Finally, to overcome the problem of detecting engravings at different distances and angles, the reference images dataset will also consider this, including reference images taken at varying distances and angles for each rock engraving. With this, we hope to achieve enough accuracy to provide the users with a good experience using the mobile application.

The solution proposed in this dissertation aims to provide the visitors of the C a Valley rock engravings additional information to complement the guided tours already available. This will be achieved through a MAR application that detects rock engravings and displays digital content in the form of text, images or animations on top of them. Furthermore, the application should be able to outline the engravings captured by the mobile device's camera to assist the user in their visualization. The user interface should be simple and intuitive to be used by all age groups.

The result of this dissertation should be a functional prototype of the application mentioned before. The prototype should be tested and evaluated in different scenarios and by different types of users.

Chapter 3

Architecture and Specification

This chapter delves into the architecture and specification of the application developed in the context of this dissertation, taking into account the research questions and the objectives proposed in sections 1.2 and 1.3. A brief description of the developed solution is presented in section 3.1. The rock art site chosen to test the application is detailed in section 3.2, followed by section 3.3, which specifies both the functional and non-functional requirements. The architecture of the developed solution is described in section 3.4. Finally, section 3.5 targets the crafting of the user experience, emphasizing on the interfaces used.

3.1 General Description

Using AR to help with the visualization of rock art can improve the experience of museum and cultural site visitors, enabling them to better understand and interpret the context and background of the art, as well as better identify specific rock engravings. Such an application should be available on mobile devices, given that many rock art sites are located in remote areas. Users should be able to use the mobile AR application to locate and outline engravings on rocks and this process should be intuitive and simple. Furthermore, being able to select and separate engravings enables new features, such as the display of animations, additional information about the selected engraving, and 3D visualizations. The application testing location and requirements were defined at the beginning of the project, although some changes were introduced to the latter during the development phase.

3.2 The Penascosa Rock Art Site

Since rock art sites are usually located in remote places, it was decided that the developed solution would target a single one, the Penascosa rock art site. Located in Vila Nova de Foz Côa, it contains 36 rocks with ancient engravings, of which five are accessible to the public. The site is located in the right bank of the river Côa and is only accessible by jeep due to the mountainous terrain between the nearest village and the site itself.

This solution focuses mostly on rock number 6 of this site, which can be seen in Figure 3.1.



Figure 3.1: Rock number 6 of the Penascosa rock art site

This rock contains four engravings, representing two horses and two goats.

3.3 Application Requirements

The definition of application requirements was a continuous process throughout the development of this dissertation. The final list of requirements was reached after an iterative process of requirement analysis carried out with the help of the client. The next two sections present the non-functional and functional requirements for the prototype.

3.3.1 Non-Functional Requirements

Table 3.1 shows the non-functional requirements of the application.

ID	Description	Priority
NF01	The mobile application should work on all Android devices with a minimum API level of 24 (Android 7.0 Nougat)	high
NF02	The device used must have a camera and the application must have permission to access it	high
NF03	The user should be able to move without losing the experience	high
NF04	Adding new information, animations, 3D models and other assets to the application should be simple	high
NF05	The codebase should be easy to extend to other rock art sites	high
NF06	The application should fully work without an internet connection	high

Table 3.1: Non-Functional Application Requirements

3.3.2 Functional Requirements

The functional requirements of the application are displayed in Table 3.2.

ID	Description	Priority
F01	Detect and track rock engravings on the scene using reference images	high
F02	Display a virtual outline of the detected engravings	high
F03	Select a specific engraving	high
F04	Display information about a specific engraving in the form of text and images	high
F05	Display 2D animations of engravings	high
F06	Color the outlined engravings	medium
F07	Detect and track horizontal planes in the scene	medium
F08	Display engraving-related 3D models on top of horizontal planes	medium
F09	Control the scale and rotation of the displayed 3D models	low

Table 3.2: Functional Application Requirements

3.4 Architecture

The architecture of the application is shown in Figure 3.2. The application launches on the Engravings Detection Page, which shows the user the input from the device's camera and detects engravings present on the received input using an Image Tracking Module and a database containing reference images. The detected engravings are then outlined and colored, making them easier to spot by the user, who then has the possibility to select specific engravings and view them separately from the rest. From there, the user can perform three actions: 1) view a page containing information regarding the selected engraving in the form of text and images; 2) view a 2D animation of the selected engraving; 3) access the 3D Visualization Page. The latter uses a Plane Tracking Module to detect horizontal planes in the real world, allowing the user to place a 3D model of the selected engraving in one of the detected planes. The model can be controlled by the user, who can change its scale and orientation. At any point in the application's execution, it is possible to return to the initial page and restart the engraving detection process.

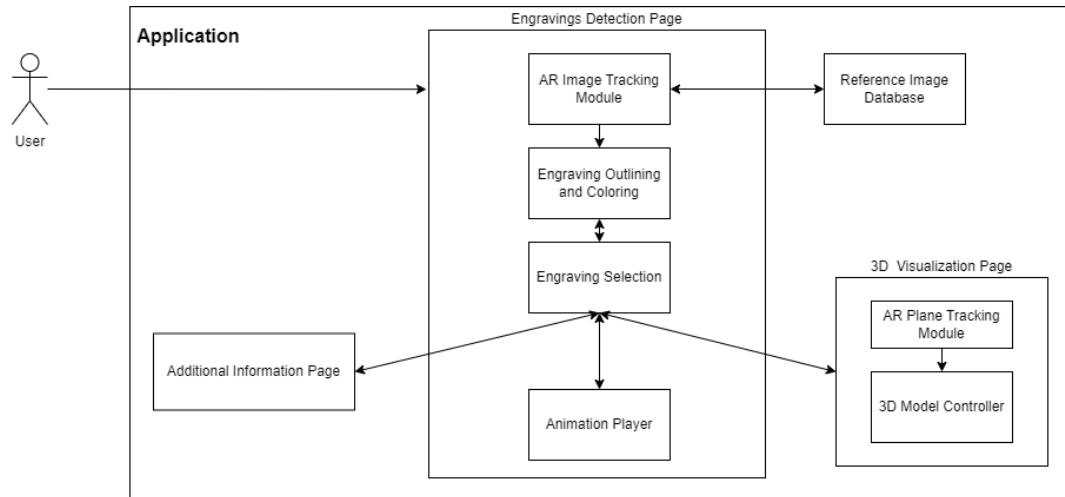


Figure 3.2: Application Architecture

The application makes use of two different AR tracking modules, one for detecting and tracking reference images and another for detecting horizontal planes in the scene. Table 3.3 shows the state of tracking modules when each application page is active.

	Image Tracking	Plane Tracking
Engravings Detection Page	Yes	No
Additional Information Page	No	No
3D Visualization Page	No	Yes

Table 3.3: Tracking states on each application page

The tracking modules are never active simultaneously due to performance and robustness reasons. However, the image tracking module keeps its previous state when disabled, being able to

recover after being suspended. Figure 3.3 shows an example of an application flow where this is demonstrated.

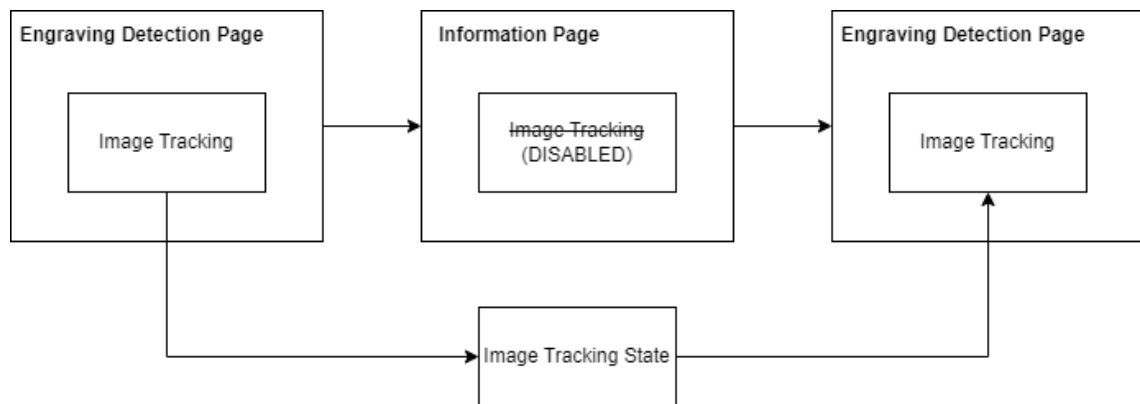


Figure 3.3: Example of an application flow where the image tracking module keeps its state

The module starts active on the Engraving Detection Page. When the user visits the Additional Information Page, the image tracking module is disabled but keeps its current state. When coming back to the Engravings Detection Page, the module recovers its previous state and continues execution from the point where it was before being suspended.

Contrarily, the plane tracking module does not store its state and cannot recover to a previous point after being disabled. The main reason is the high computational overload of storing all the objects related to each detected plane.

3.5 User Experience

3.5.1 User Interfaces

3.5.1.1 Outline Selection

After the engravings have been detected in the camera view, they are tracked by the AR tracking module and virtual outlines of each detected engraving are shown to the user. Figure 3.4 shows an example of how rock engravings look with and without virtual outlines on top.

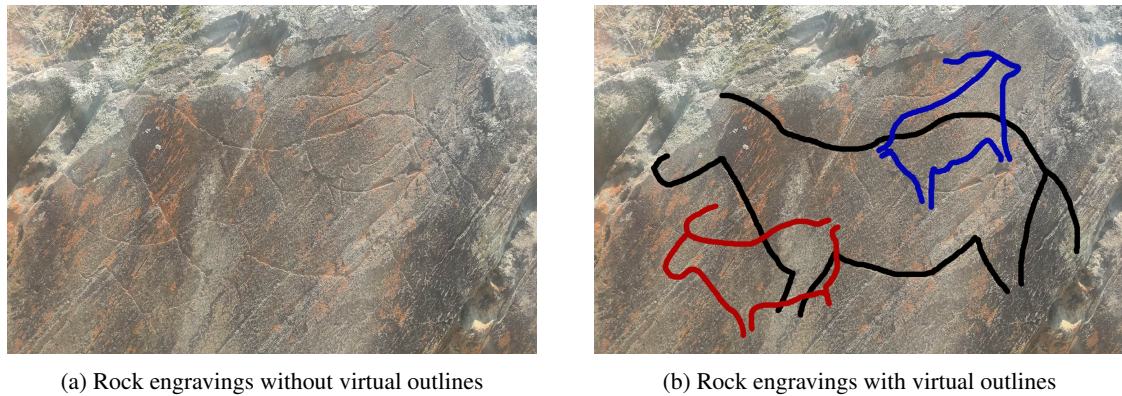


Figure 3.4: Rock engravings without (a) and with (b) virtual outlines

In order for the user to select a specific outline to view separately, two options were considered: 1) the user would simply tap the outline he wished to select; 2) a panel containing a small picture of each engraving would be displayed on the screen, as shown in Figure 3.5, allowing the user to select a specific engraving from there.



Figure 3.5: Engraving selection interface

After testing both options, it was concluded that the first one was not viable. Firstly, since the outlines are mostly thin lines, the user would have to tap the screen with a high degree of precision to hit the desired outline, negatively contributing to the overall user experience. Furthermore, in the case of having overlapping engravings, as is the case of Figure 3.5, there were problems identifying which outline the user was trying to select in case the screen tap happened at a junction where two different outlines met. Because of these reasons, the second option was preferred. Using this option, the user is presented with a small panel on the bottom of the screen that contains a button for each engraving detected. Users can select a specific engraving by tapping the corresponding button in the panel.

3.5.1.2 Engraving Information

The user is able to view information regarding a specific engraving after selecting it, in the form of text and images. Two options for displaying this information on the screen were considered: 1) displaying the information in the AR scene, close to the detected engraving, appearing as if the text and images were placed in the real world, as shown in Figure 3.6; 2) displaying the information on a different page of the application.

The first option presented several problems, namely regarding the orientation of the device's camera. As the text and images were placed in the AR scene, the movement of the mobile device's camera affected its orientation, making it difficult to read text or analyze images in some cases. Figure 3.6 shows how the information would be displayed on the screen according to the first option.



Figure 3.6: Example of displaying text in the AR scene, next to the engraving

The second option, shown in Figure 3.7, presented a better alternative, allowing the user to visualize all the information about an engraving on a single page. Furthermore, this option also allows for better control of how the information is organized on the screen and makes it easier to add new information.

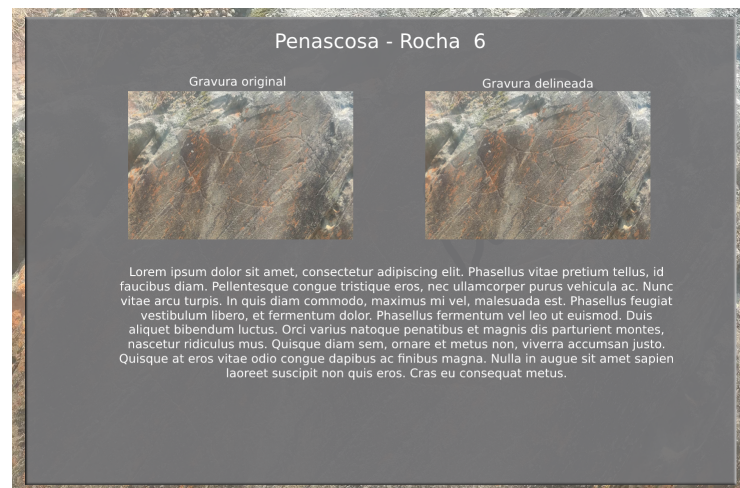


Figure 3.7: Displaying information about an engraving on a different page

After testing both options, the second one was considered best for the user experience and was, therefore, chosen to be implemented in the final solution.

3.5.1.3 3D Model Control

One of the aspects that enhances the user experience is the ability to control the scale, rotation and position of the 3D models spawned in the scene. After selecting a specific engraving, the user can choose to view a 3D representation of what is depicted in the rock. After starting the 3D visualization process, the AR module detects the available horizontal planes in the scene captured by the camera and the user can spawn a 3D representation of the engraving by tapping one of the detected planes. Depending on the user's location, the model's original scale might have an inappropriate size. It is, therefore, important that the scale can be controlled while using the application. Furthermore, controlling the rotation of the spawned model is also critical in improving the user experience.

Regarding the control of the model's scale, two design options were explored: 1) having a slider on the screen that allows the user to increase or decrease the size of the model; 2) having three predefined scales that the user can choose from.

The first option, shown in Figure 3.8, allowed greater control over the model's scale and a higher degree of customization. However, the slider occupied a significant portion of the screen.



Figure 3.8: Slider that allows the user to control the scale of the 3D model: real-size model (a); bigger model (b)

The second option includes three buttons on the side of the screen, each corresponding to a different predefined scale for the model, as shown in Figure 3.9. Although this option restricts the degree of control over the model's scale, it occupies significantly less screen space than the first option, leaving more room for the input captured by the camera.

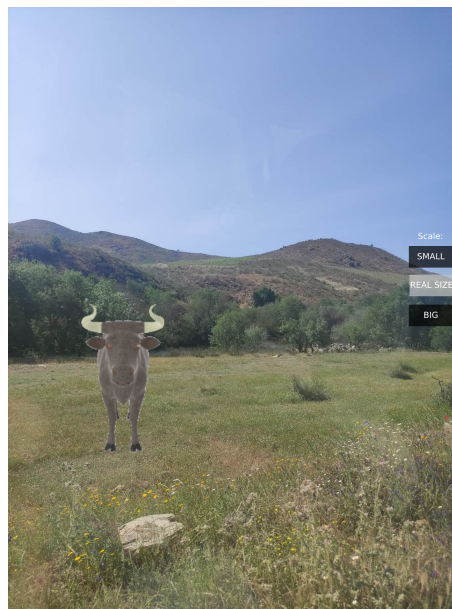


Figure 3.9: Predefined scales for the 3D model

The first option was considered better for this application as it provides a higher level of control to the user, allowing a wider range of scale values to be applied.

Regarding the model's rotation, two design options were explored: 1) having a slider on the screen that allows the user to rotate the model freely; 2) having the model always face the camera.

Including a slider on the interface, takes a significant amount of screen space but provides a higher degree of control. As shown in Figure 3.10, the user would be able to rotate the model by moving the slider.



(a) Rotation slider showing a 3D model facing the camera



(b) Rotation slider showing a 3D model at a side angle

Figure 3.10: Slider that allows the user to control the rotation of the 3D model: front-facing model (a); side-facing model (b)

The second option has no impact on the interface, as it does not need any UI elements. However, having the model always facing the camera introduced a major problem, only allowing the user to visualize its front side. Because of this, the first option was preferred.

Chapter 4

Methodology/Solution

This chapter describes the methodology and the solution specified in Chapter 3, with the main objective of enriching the experience of rock art site visitors, allowing them to better identify and understand rock engravings.

4.1 Tools and Technologies

4.1.1 Development Environment

One of the first stages of the application development was deciding which tools and technologies would be used to build the prototype. This decision could significantly impact both the development process and the quality of the final solution. Because of this, various technologies were explored and compared before reaching a final decision. Since the solution consists of a mobile AR application, the ability to deploy and test it on a mobile device quickly was a critical aspect that needed to be met. For this, there were two main options: 1) developing natively for Android and iOS; 2) using an external engine, like Unity or Unreal. Option 1) has the benefit of giving the developer full control over the AR system but presents an increased level of complexity. Furthermore, there would be the need to build two different applications, one for Android and one for iOS, since the same code cannot be applied to both operating systems. Therefore, the second option was chosen. Using an external engine provides a higher level of abstraction to the programmer regarding the AR features and enables quick deployments for both operating systems without requiring code changes. Unity was chosen out of the available engines since it works with the majority of AR frameworks, has rich documentation, and a sizable community.

4.1.2 AR Framework

The second technological choice for the project concerned the AR platform to be used. Since it is the backbone of the application, this decision involved testing different frameworks and comparing their features and performance. This section describes the process of choosing which AR

framework to use among the two that were considered: Vuforia¹ and ARFoundation². Both are simple to integrate with Unity and include useful features for the developed solution, like Image Tracking and Plane Tracking.

For the purpose of testing and comparing both frameworks, a small prototype was developed, consisting of a simple mobile application able to detect and track reference images. The objective of this prototype was to compare the image detection and tracking features of both frameworks when used on flat rock engravings. The tests were conducted with two reference images, each containing a different engraving from the same rock, as shown in Figure 4.1.



Figure 4.1: Reference images used in the initial prototype

4.1.2.1 Vuforia

Vuforia contains both image and plane tracking features and allows applications to be deployed to Android and iOS.

Vuforia's image tracking module uses Image Target Manager objects as its basis. Each of these objects can detect one reference image in the input captured by the camera. Although this framework successfully detected and tracked the reference image with high accuracy, it had the drawback of requiring the creation of an Image Target Manager object for each. Since the solution requires multiple reference images for each rock and the number of images is not the same for each one, Vuforia's regular image tracking module is not a viable solution. Although the framework provides an alternative for this, the Cloud Recognition feature, able to match the camera input with all the reference images contained in a database stored in the Vuforia Cloud servers, this feature requires internet access. Due to the remote locations of the Côa Valley rock art sites, mobile phone reception is frequently nonexistent, making internet access similarly nonexistent. As a result, it was determined that the Vuforia AR framework did not satisfy the project's criteria.

¹Vuforia Homepage: <https://developer.vuforia.com>

²ARFoundation package homepage: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>

4.1.2.2 ARFoundation

ARFoundation is a package that allows developers to create multi-platform AR apps and, as Vuforia, incorporates both image and plane tracking features. By using ARCore³ and ARKit⁴ as its underlying AR engines, applications built with ARFoundation can easily be deployed to Android and iOS devices.

ARFoundation contains an Image Tracking module that is capable of detecting multiple images at once from a database of reference images. This has significant benefits for the proposed AR system because it eliminates the need to create new objects for each reference image. The Image Tracking module, after populating the reference database, compares the features found in each reference image to the ones found in the camera input and, in the event of a match, tracks the detected image. The reference database is stored in the application's memory, therefore allowing it to properly work offline. The increased difficulty of debugging jobs is one of ARFoundation's key downsides. While Vuforia allows the developer to simulate a mobile device's camera by using the one from the computer, which enables the application to be executed in the Unity testing and debugging environment, ARFoundation does not support this feature, requiring the application to be built and deployed to a mobile device in order to be tested. Since the process of deploying the application usually takes a few minutes, this is a very time-consuming problem. Still, ARFoundation was chosen as the preferred AR framework for this project due to the versatility it offers in image detection and tracking.

Table 4.1 shows a comparison of the most important features for this solution that are offered by each of the explored AR frameworks.

	Vuforia	ARFoundation
Android Build	Yes	Yes
iOS Build	Yes	Yes
Image Tracking	Yes	Yes
Plane Tracking	Yes	Yes
Offline multiple image tracking	No	Yes
Testing in the Unity environment	Yes	No

Table 4.1: Features comparison between Vuforia and ARFoundation

4.1.3 Image Editing

Due to the fact that the outlines had to be created separately for each reference image, creating this solution required extensive image editing work. Two technologies were used for this matter:

³ARCore Homepage: <https://developers.google.com/ar>

⁴ARKit Homepage: <https://developer.apple.com/augmented-reality/arkit/>

1) Adobe Photoshop ⁵ and 2) Photopea ⁶. The main use of these technologies was, as mentioned earlier, drawing and exporting the engraving outlines, as shown in Figure 4.2.



(a) Original rock engraving



(b) Outline extracted from the engraving

Figure 4.2: Original engraving (a) and outline extracted from it (b)

Apart from this, Photoshop and Photopea were also used to sketch some of the application interfaces and to create reference images with altered contrast and brightness values.

Another use of these tools was the artificial creation of new reference images. Based on pictures of the target rock or engraving, new reference images were created by altering the pictures' contrast and brightness values, as shown in Figure 4.3.

⁵Adobe Photoshop Homepage: <https://www.adobe.com/products/photoshop.html>

⁶Photopea Homepage: <https://www.photopea.com/>



(a) Original reference image



(b) High brightness reference image



(c) Low brightness reference image



(d) High contrast reference image



(e) Low contrast reference image

Figure 4.3: Reference images artificially generated from an original picture

4.2 UI Design

This section describes the user interfaces (UI) implemented in the application, detailing some elements. Although portrait mode is the most used in mobile AR applications, the developed solution is locked into landscape mode. It was decided that this would be the best orientation for the solution due to the better results presented in the preliminary image detection and tracking results compared to portrait mode. Given that most engravings are displayed horizontally and that reference images need to occupy a certain percentage of the screen to be detected, portrait mode

proved ineffective since it was difficult to fit most engravings in the screen with a big enough size to be recognized and detected.

4.2.1 Initial Page

The initial page of the application is the first page a user sees and should, therefore, be simple and captivating. Figure 4.4 shows the user interface implemented for this page.



Figure 4.4: User Interface for the initial page

The implemented UI is simple, containing a background image alluding to the engravings, the symbol of the Côa Museum, a title and a subtitle. It also contains a button to start the execution of the application, which leads the user to the Engraving Detection Page.

4.2.2 Engraving Detection Page

On this page, the user can detect engravings captured by the device's camera. Therefore, the input the camera captures is shown as the background so the user can clearly visualize what it is capturing. Furthermore, this page contains different elements depending on which state it is in. There are three possible states for the page: 1) no engravings have been detected, 2) engravings have been detected and their virtual outlines placed on the scene and 3) a specific engraving has been selected to be isolated from the others. Figure 4.5 shows the UI for the first state, containing only the camera input as background and a panel across the bottom of the screen.

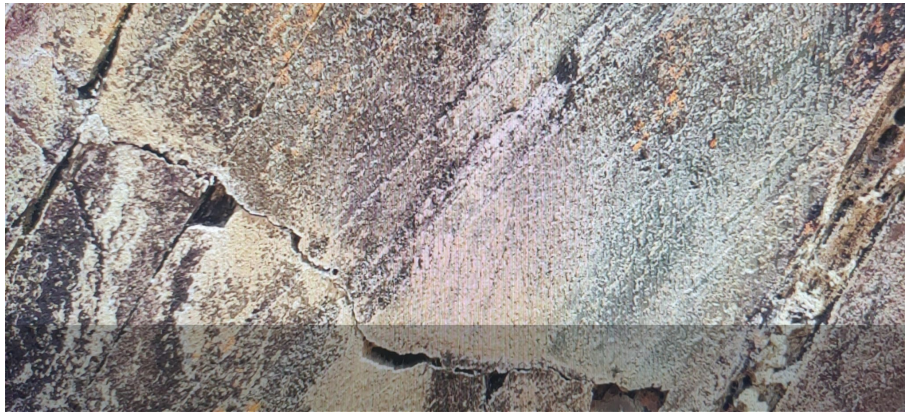


Figure 4.5: User Interface for the Engraving Detection Page before any engraving is detected

After detecting engravings, their virtual outlines are placed on the scene and colored. Furthermore, a button containing the engraving's outline is added to the panel at the bottom of the screen for each detected engraving. These buttons can be used to select a specific engraving from the ones available. Figure 4.6 displays the UI for the second state of the page.

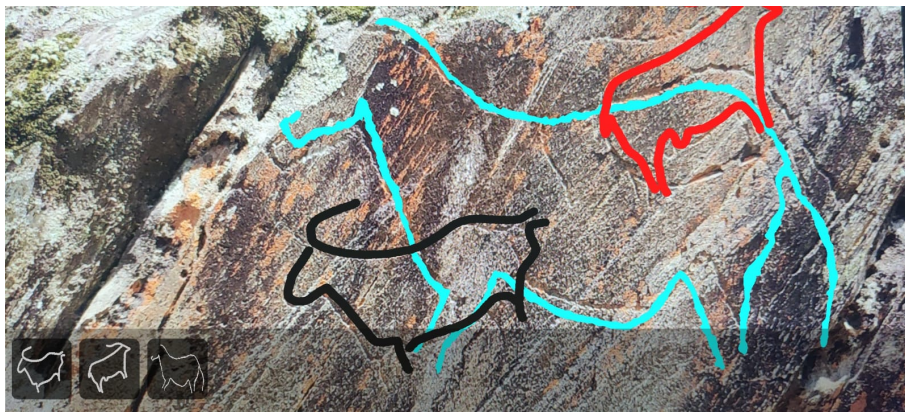


Figure 4.6: User Interface for the Engraving Detection Page after engravings having been detected

Finally, after the user selects an engraving, only that engraving's virtual outline is shown, with the others being hidden. Moreover, three buttons appear on the right side of the screen, each with a different function. Due to most users preferring to use their right hand to interact with the device, the action buttons are all placed on the right side of the screen. Figure 4.7 demonstrates the UI implemented for the third state of this page.



Figure 4.7: User Interface for the Engraving Detection Page after a specific engraving has been selected

4.2.3 Additional Information Page

This page displays information regarding the rock or engraving the user has selected to view. As shown in Figure 4.8, its UI is divided into two parts. The left part of the screen contains a textual description of the rock/engraving, along with a button to return to the last page. The right part of the screen contains images of the rock/engraving in a slideshow.

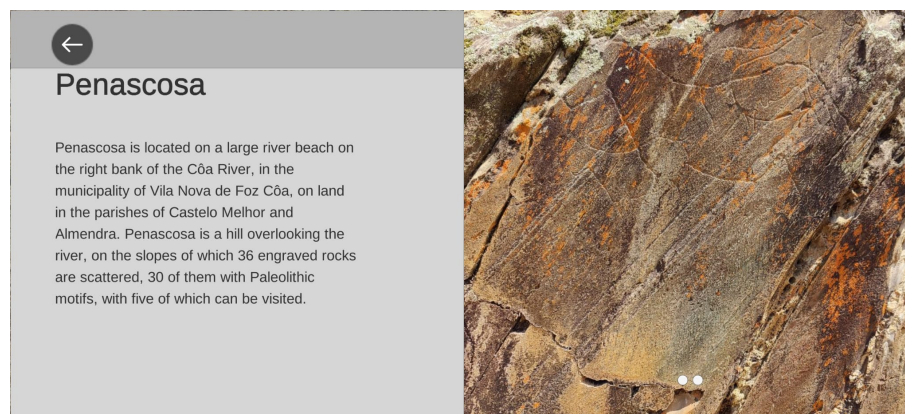


Figure 4.8: User Interface for the Additional Information Page

4.2.4 3D Visualization Page

The user can visualize a 3D representation of an engraving on this page. When entering it, a hint is shown in the middle of the screen, prompting the user to point the camera to the ground, as illustrated in Figure 4.9. The hint is removed from the screen when a plane is detected.



Figure 4.9: User Interface for the 3D Visualization Page before planes having been detected

The detected planes are displayed on screen using orange polygons and the user can tap them to spawn a 3D model of the previously selected engraving. Furthermore, the two sliders at the bottom of the page can be used to control the model. Figure 4.10 shows the UI for this page.



Figure 4.10: User Interface for the 3D Visualization Page after planes having been detected

4.3 Image Detection and Tracking

The most important step of the application's execution is the detection and tracking of rock engravings. This section describes how this part of the application was implemented, focusing on Android devices and, therefore, ARFoundation's Android AR subsystem, ARCore.

ARFoundation allows the creation of a reference database that contains all the reference images to be searched in the received camera input. When creating the reference database, ARCore's computer vision algorithms are used to extract features from each reference image and the extracted features are stored in one or more feature databases. When executing the application, the AR engine performs a search for features in the camera input that match the ones stored in the feature databases and, if a match occurs, the matched image is then tracked. Figure 4.11 illustrates this process.

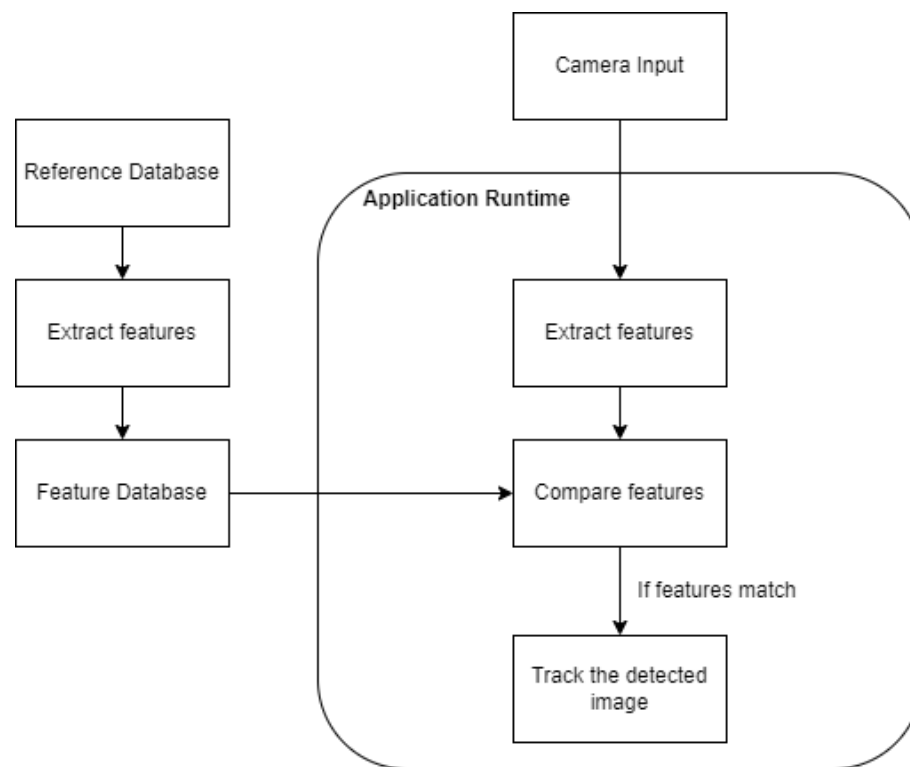


Figure 4.11: Image detection and tracking process

ARCore provides a tool called *arcoring*⁷ that analyses and evaluates reference images, giving them a quality score from 0 to 100. The scores provided by this tool were used as a guide to determine the quality of reference images.

4.3.1 Approach

Two approaches were implemented and tested for this issue: 1) detecting each rock engraving separately; 2) detecting the entire rock. Each approach was incorporated into a different application, therefore resulting in two different versions of the proposed mobile application. All features were kept the same between both versions except for the tracking approach.

4.3.1.1 Detecting engravings separately

The premise of this option was to allow users to freely detect engravings without needing to fit the entire rock in the camera view, which, a priori, seems like a more user-friendly and intuitive approach. This approach consists of using multiple reference images specific to each engraving, which are used to detect them separately. This means that if the target rock is captured partially by the camera, only the engravings fully present in the visible part are detected. As demonstrated in Figure 4.12, only one of the engravings is detected in this approach since it is the only one that

⁷ARCore's *arcoring* tool: <https://developers.google.com/ar/develop/augmented-images/arcoring>

fully appears on the camera view. Although the camera partially captures other engravings, the application is not able to detect them.

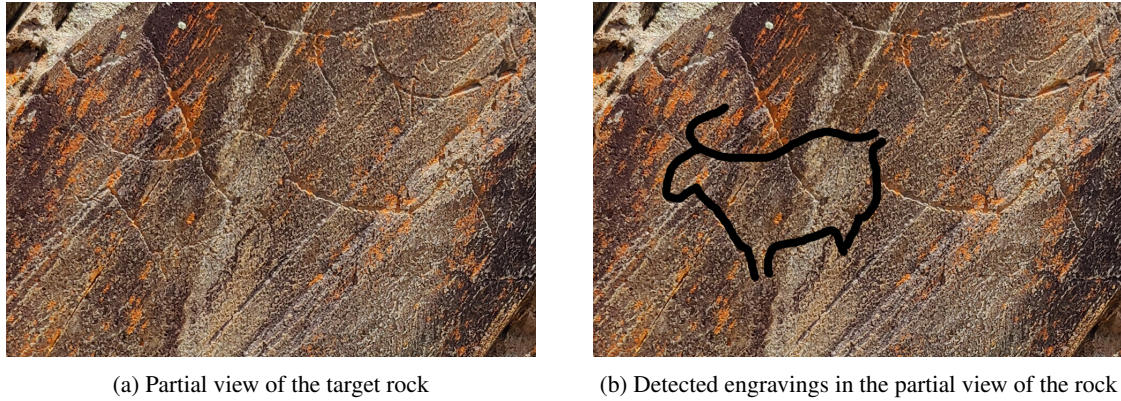


Figure 4.12: Target rock captured only partially (a) and the engravings detected (b)

After extensively testing this approach, some problems were discovered. Firstly, detecting engravings separately impacted the fluidity of the application. Since some engravings are harder to detect than others due to being smaller or having thinner lines, they would take longer to be recognized. This led to the application not detecting all the engravings present in the camera view at the same time. Furthermore, due to the quality of each engraving's reference images, some required the user to get closer or move the camera around in order to be detected, while others did not. The differences in engraving size also posed a significant problem. Larger ones required the user to stand at a greater distance from the rock than smaller engravings, which introduced inconsistencies in their tracking. After detecting a large engraving, the user needs to get closer to the rock to detect a smaller one, sometimes leading to the larger engraving leaving the camera view either completely or partially, resulting in the loss of its tracking.

Moreover, this approach is significantly more computationally expensive than the alternative, as it requires a greater number of reference images to be included in the database. Since engravings are detected on their own, each engraving requires its own set of reference images, including pictures of the engraving at different angles and with different brightness and contrast levels. With n being the number of reference images needed for a single object to be detected and x the number of engravings comprised in the target rock, table 4.2 shows the difference in the number of reference images needed by each tested approach.

	Separate engravings detection	Rock detection
Reference images needed	$x * n$	n

Table 4.2: Number of reference images needed by the different image detection approaches tested, with n being the number of images needed for the detection of a single object and x the number of engravings in the target rock

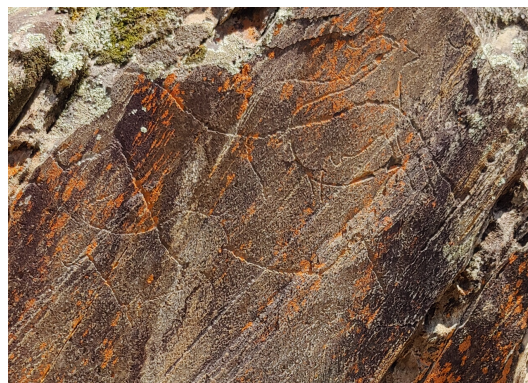
The number of reference images needed increases linearly with the number of engravings contained in the target rock. This is not an issue for rocks with a small number of engravings but becomes a significant problem for those that contain a high number of them.

Still on the performance side, this approach also implied the tracking of multiple images at once, requiring more resources to estimate the pose of each tracked object relative to the camera, which increased the number of calculations performed between each frame, therefore negatively impacting the frame rate of the application. Overall, this approach achieved worse performance results than its alternative.

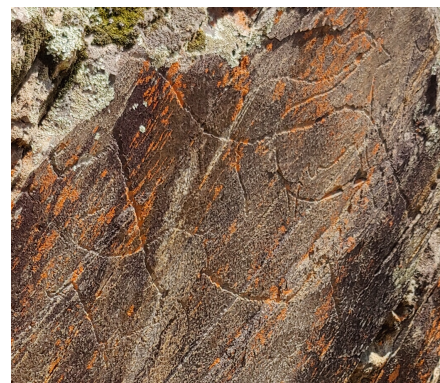
4.3.1.2 Detecting the entire rock

Detecting the entire rock was the easiest approach to implement, as the position of each engraving in the rock is known a priori. After detecting it, the virtual outlines for each engraving were instantiated in their position relative to the original reference image used in the detection phase. This option requires fewer reference pictures than its alternative, improving the performance of the application due to the smaller number of feature set comparisons required.

At first, the reference images used contained the entire target rock and each displayed a different angle of it, as shown in Figure 4.13.



(a) Reference image containing the entire rock from a front view



(b) Reference image containing the entire rock from a leftside view

Figure 4.13: Reference images containing the entire rock, at different angles

These images usually achieved higher quality scores than images of specific engravings due to having a higher amount of feature points. However, since full rock images often contained elements that were not part of the rock itself, like vegetation, they were harder to use in real-world scenarios. Furthermore, the higher amount of features meant that matching one of these reference images was more difficult due to the feature sets collected from them being more specific and complex. Generally, reference images that contained the entire rock were harder to detect than ones containing single engravings.

In order to overcome this issue, single engraving images started being used as reference instead of full-rock ones. Since the location of each engraving relative to all the others is well known, one

only needs to locate a single engraving to deduce the locations of the others. Therefore, a specific engraving from the target rock was chosen to be used as the target for image detection. In this case, for rock number six of the Penascosa rock art site, the chosen engraving was the one representing a horse in the middle of the rock, as illustrated in Figure 4.14.



Figure 4.14: Image containing the engraving to be used as reference

After detecting this image in the camera view, the virtual outlines are placed in the positions of their corresponding engraving relative to the one detected. This change decreased the difficulty of detecting the target rock, which, in turn, improved the user experience.

This approach also proved to be significantly less expensive computationally, as mentioned before in section 4.3.1.1, requiring fewer reference images than the alternative option and fewer resources to track objects. Since only one object is being tracked (the entire rock or the reference engraving), the pose estimation calculations are only performed once, for the tracked object, instead of multiple times for each tracked engraving.

4.4 3D Visualization

After selecting a specific engraving from the ones detected by the application, the user can choose to visualize a 3D representation of it, if available. By choosing a horizontal surface present in the real world, it is possible to instantiate and control a 3D model corresponding to the animal depicted in the engraving.

The first step of this feature is the detection of horizontal surfaces where the model can be placed. During this phase, the application disables image tracking and enables the plane tracking module. Horizontal planes captured by the camera are detected and their surface is rendered orange on-screen.

After detecting at least one horizontal surface, the user can place the model on the scene by simply tapping the detected plane. The model is instantiated at the tap location and its position can be changed by performing another tap on a valid horizontal surface. Moreover, the user can freely rotate the model using a slider at the bottom of the screen. This ensured that all sides of the spawned object are visible without the user needing to move around in the real world to see

them. The size of the model can also be controlled. The model is originally instantiated at a real scale, having the same size it would have if it were a real object. The user can then use another slider to either increase or decrease the size of the 3D object. This is especially useful in situations where, for example, the user places the model close to the device and the model's size needs to be decreased.

Finally, the models are animated, which increases their realism and integration into the real-world scene.

4.5 Implementation Details

4.5.1 Virtual Outlines

Virtual outlines are placed on top of the detected rock engravings to help the user identify them in the real world since most engravings consist of very thin lines hard to locate and distinguish without perfect lighting conditions. In some rocks that have multiple overlapped engravings, virtual outlines can help locate the lines but distinguishing between engravings still remains a difficult task to the user. Therefore, virtual outlines can also be colored, as shown in Figure 4.15.



Figure 4.15: Colored virtual outlines

When the target rock is detected in the camera input, a plane object is instantiated for each engraving contained in the rock. The planes are positioned on top of their respective engraving and the outline is rendered onto the plane's mesh. Outlines possess a transparent background to allow the overlapping of multiple planes and also so that the rock does not get covered by the outline's background. A new Unity material is created for each outline, containing its texture and using the Cutout rendering mode. The materials are then applied to their respective plane, resulting in the outline appearing on the scene, on top of the original engraving.

4.5.2 Rock Information

Information regarding each rock is stored in the device's memory so that it can be displayed when the user visits the Information page. for each rock, a JSON⁸ file is stored. Each file contains the

⁸JSON homepage: <https://www.json.org/json-en.html>

following information:

- *string* - The title of the rock
- *string* - A textual description of its context and background story
- *int* - The number of engravings contained in the rock

For each engraving, another JSON file is stored, with the following fields:

- *boolean* - Whether the engraving has an animation or not
- *int* - The number of animation frames
- *string* - The color of the engraving

These fields are used to correctly instantiate the engraving's virtual outline, including its animation.

4.5.3 2D Animations

After detecting engravings and selecting a specific one, the user can play a 2D animation on the detected engraving's virtual outline. In order to maximize efficiency, these animations are texture-based and the outline image contains all the animation frames displayed sequentially, as shown in Figure 4.16.



Figure 4.16: Outline texture for the horse engraving

The number of animation frames contained in the image is stored as a parameter in the JSON file relative to the engraving. When rendering the virtual outline's texture onto its respective plane, the scale is adjusted so that only the first frame is rendered on the plane.

The texture scale, for an arbitrary virtual outline, is the following, where *numFrames* is the number of animation frames contained in the outline's texture:

$$x = \frac{1}{numFrames}$$

$$y = 1$$

For the outline texture shown in Figure 4.17, containing a simple animation for the horse engraving with three frames, so that only the first frame is rendered onto the plane, the texture scale is the following:

$$x = \frac{1}{numFrames} = \frac{1}{3} = 0.333$$

$$y = 1$$

This restricts the part of the image that is rendered on the plane, ensuring that only the first frame is displayed.

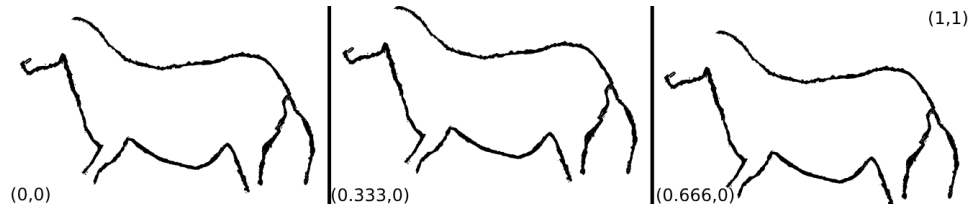


Figure 4.17: Outline texture for the horse engraving with texture coordinates

Animations consist of offsetting the texture being rendered on the plane's mesh on an interval basis. Keeping the scale of the previous example, Table 4.3 shows the offsets applied to the texture's x axis to display each frame.

Frame	Offset
1	0
2	0.333
3	0.666

Table 4.3: Texture offsets used to display each frame of an animations containing 3 frames

Applying these offsets at predetermined time intervals results in the 2D animation being played. These animations are displayed on top of the detected engraving, therefore giving the user the sense that it is being animated in the real world.

Chapter 5

Performance Evaluation

Evaluation of the developed solution focused on two areas: 1) performance and 2) usability. This chapter describes the performance evaluation step starting, in section 5.1 with a description of the different testing environments used to evaluate the application's performance. This is followed by a presentation of the techniques explored to improve the image detection step and the tracking of already detected images, in section 5.2.

5.1 Testing

This section describes the testing process used for this application.

5.1.1 On-site Testing

On-site tests are the most reliable way of validating the developed mobile application. Since the environment where the tests are conducted is the same as the one where the application will be used, these tests reflect the application's behavior in real conditions. However, due to the Penascosa rock art site being a protected environment, visits to the target rock had to be done with permission from the Côa Museum and accompanied by a guide, rendering the frequent visit to the Penascosa site for testing purposes unviable.

Although on-site testing produces the most accurate results, changing the application's code during the test session is a complex process. The most critical obstacles encountered in this type of testing were the unavailability of mobile phone coverage and the fact that the site is not equipped with a power source.

Overall, on-site testing was impractical but a critical part of the development process given that the results produced in these tests provided great feedback.

5.1.2 Simulated Testing

Most tests were carried out in a simulated environment due to the remote location of the Penascosa rock art site and the unfavorable conditions on it. Testing on a simulated environment allows tests

to be performed with a high frequency and changes to be implemented quickly. Nevertheless, due to the dissimilarity between the testing environment and the actual application usage scenario, caution was necessary when interpreting these tests' outcomes.

Various pictures of the target rock were taken at different angles and distances to simulate real-world conditions. These pictures were then separated into two groups, one to be used as reference images for the application, and the other to be used for testing purposes. Ensuring that the application is tested using pictures that are not the same as the reference images was a critical step in the testing process, contributing to the validity of the test results.

Two main ways were used to test the detection of engravings: 1) using printed pictures and 2) using a computer to display the pictures. The first way allowed for the application to be tested in different lighting conditions since the printed pictures could be taken to different locations, with different lighting settings. The second method was easier to set up than the first, requiring only a computer, usually the one where the application was also being developed, to display the pictures. However, it was found that the detection and tracking of engravings was sometimes influenced by the reflections of the computer's screen. Still, this was the main method used for testing due to the significantly easier setup and the testing environment being ready much faster using this technique.

5.2 Performance Evaluation - Image Detection and Tracking Improvements

In order to improve the detection and tracking of rock engravings, some ideas were explored and tested. There are two critical steps for this application, regarding this point:

1. The initial detection of a reference image.
2. The tracking of the detected image's position in relation to the mobile device.

The first step's performance is evaluated by measuring the time it takes for the AR system to detect a reference image when it is indeed being captured by the camera. This metric will be referred to as **Metric 1.a**. Furthermore, step 1 is also evaluated by measuring the robustness of the image detection module. Given that the application will be used outdoors, it needs to be able to match as many different views of the target rock as possible. This metric will be referred to as **Metric 1.b**. This step is tested manually and the evaluation metrics used are subjective, since there is no method to concretely determine when this step begins. Therefore, performance improvements and reductions are measured from the feedback gathered by the users who tested the application.

The second step's performance can be measured in multiple ways. Firstly, the number of frames processed per second by the application, referred to as **Metric 2.a**, can provide significant insights into the performance of the image tracking process, after a reference image is detected. This is the only concrete performance metric. Apart from it, the accuracy of the position estimated

by the AR engine relative to the actual position of the engraving (**Metric 2.b**) and the number of times the tracking of the reference image is lost while the picture is still on the screen (**Metric 2.c**) are also measured. Although these are subjective metrics, gathered from the users' feedback, they can provide important information regarding the performance of this step.

All the approaches to improve these two steps were compared to a baseline version of the application that used a single reference image to detect the target rock and the raw input captured by the mobile device's camera.

This section describes the tests performed and the results achieved.

5.2.1 Multiple reference images for each engraving/rock

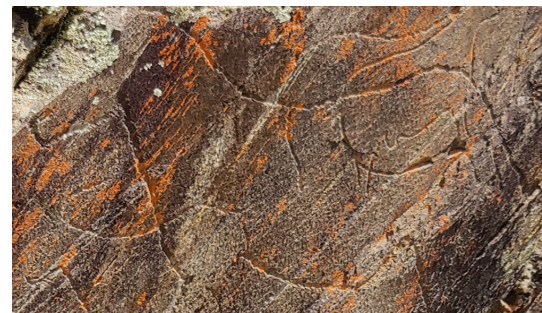
The first idea that was tested was the use of multiple reference images for each tracked object. This idea focused on improving both steps (1 and 2). The variations introduced in the reference images targeted different aspects, including the angle and distance at which the image is visualized, as well as its brightness and contrast values. An object's reference database included at least three pictures taken at different angles (one from a frontal view, and one from each side), as shown in Figure 5.1. Furthermore, since the pictures were taken on the actual rock art site, they only include angles from which it is physically possible to view the target rock, hence why some of the reference images are angled in slightly different ways vertically.



(a) Frontal view image



(b) Right-side view image



(c) Left-side view image

Figure 5.1: Reference images from different angles: front (a), right (b), left (c)

For each of these, four more reference images were used, with different values of brightness and contrast to try to simulate different outdoor lighting conditions. Although these artificially

generated image versions did not always provide significant improvements to the quality of the original reference image, they still could highlight features of the targeted engraving that were not discernible on the original picture. Table 5.1 shows the quality scores obtained by each version of the reference images, calculated using the *arcoring* tool.

	Baseline (original image)	Brightness		Contrast	
		High	Low	High	Low
Frontal View	0	0	20	45	0
Left-side View	20	15	0	0	0
Right-side View	0	0	0	15	0

Table 5.1: Quality scores obtained by each artificially generated reference image, calculated with the *arcoring* tool

One important aspect to notice is the existence of various reference images with a score of 0. Given that the *arcoring* tool tends to give poor quality scores to images with repeating geometric patterns, some of the pictures that contain parallel lines resulting from the rock’s shape itself or similar vegetation patterns obtained very low scores. However, they were still trackable and the application was still able to detect them. As shown in Table 5.1, the results were not consistent across all viewing angles. While for the left-side view, the original image had the highest quality score, the high-contrast version proved to improve the reference image quality for both the frontal and the right-side views.

One of the problems encountered early on while developing this improvement was that the application could detect more than one reference image at the same time. For example, the application could detect the frontal view baseline image and also the high-contrast version of that same view. In this case, the virtual outlines were instantiated twice and the performance significantly decreased. To fix it, all the reference images related to the same engraving viewing angle were given a name with the same prefix that acts as an identifier. This way, when the application detects a reference image, all the other reference pictures containing the same prefix are blocked from being detected and tracked, ensuring that each engraving’s virtual outline is only instantiated once and, therefore, saving computational resources. Given that this improvement to the tracking increases the computational cost of the image detection feature, since the features detected on the camera input have to be compared to more reference images than if only the baseline image was used, this saving of resources proved to be critical.

Apart from the additional reference images, this approach also requires a specific version of the engraving outlines for each viewing angle. Since the shape of the engravings changes with the angle at which they are detected, if the original outlines were used for all viewing angles, they would mismatch the actual engraving lines on the real-world rock. As seen in Figure 5.2, the outline used for the frontal view does not exactly match the engraving lines for the right-side view. Therefore, different versions of the original outlines had to be created for each viewing angle of the original image.

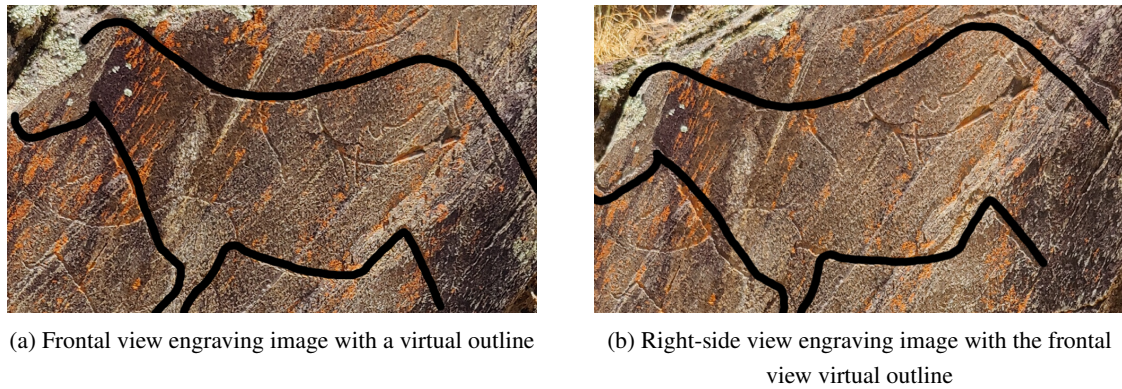


Figure 5.2: Different viewing angles using the same outlines

As shown in Table 5.2, users who tested this approach reported a decrease in performance when considering the time the application took to detect a reference image (Metric 1.a), when compared to the baseline approach. By using more reference images, more comparisons between feature sets have to be made each time, therefore increasing the computational complexity of the application. Although this can be a problem, this method was able to match more images of the target rock, different from the one used as a reference, increasing the robustness of the image detection module. The number of frames processed per second did not suffer any changes with this approach, as demonstrated in Table 5.3, with metric 2.a remaining at the maximum possible value for this application of 30 frames per second. Moreover, users did not report any significant changes regarding metrics 2.b and 2.c.

Metric	Result
1.a	Worsened
1.b	Improved
2.a	No change
2.b	No change
2.c	No change

Table 5.2: Performance metrics results for the approach of using multiple reference images for each engraving/rock, including 3 different angles and 5 different pictures for each angle

Approach	FPS
Baseline	30
Multiple reference images	30

Table 5.3: Number of frames processed per second by the application using each approach

This technique worsens the time it takes to detect a reference image but highly increases the robustness of the image detection module. In order to minimize the decrease in performance when

detecting reference images while achieving high robustness, different sets of reference pictures were tested. The set that produced the best results amongst the users was one containing only three reference pictures for each angle: the original one, one with high contrast and one with high brightness. Regarding the angles, since Penascosa's rock number 6 is difficult to visualize from the right side due to the terrain being uneven, this angle was not used to produce reference images. The final tests were carried out using six pictures in total, three for each angle, with two different angles. Table 5.4 shows the results obtained in these tests. Due to the smaller number of reference images being used, the higher computational complexity was not significant enough to affect the users' perception of the application performance. The robustness of the image detection module was still increased, with this version of the application being able to match more pictures of the target rock than the baseline version.

Metric	Result
1.a	No change
1.b	Improved
2.a	No change
2.b	No change
2.c	No change

Table 5.4: Performance metrics results for the approach of using multiple reference images for each engraving/rock, including 2 different angles and 3 different pictures for each angle

Overall, this approach increased the complexity of the application with the addition of different outline versions and the need to block the detection of reference images that are related to an already detected object. Nevertheless, it brought essential improvements to the robustness of engravings detection in the real world, which is the most critical part impacting the user experience. Step 2 was not affected by this approach. Due to these reasons, this idea was implemented in the final version of the application.

5.2.2 Removing the background

One of the main problems affecting the rock detection was that the reference images usually contained excessive elements, like other background rocks or vegetation. This, in turn, leads to the images having more features extracted from them, which is not useful when trying to detect a specific rock or engraving. Since the image detection algorithm also uses the features generated from the other elements, these increase the complexity of the image tracking module. Furthermore, elements like vegetation could produce features that are not representative of the real world anymore since these elements are unpredictable and the state of the vegetation at the time of capturing the reference image can be, and probably is, different than the one captured at runtime. As shown in Figure 5.3, to capture the entire rock or even some specific engravings in a single picture, additional elements were usually also captured.



Figure 5.3: Reference image containing vegetation elements (top-left corner) and parts from other rocks (bottom-right corner)

The idea of removing unwanted elements from the reference image's background arose as an option to solve this issue. Figure 5.4 compares an original reference image to its alternative version, with the useless elements removed.



(a) Original reference image



(b) Reference image with unwanted elements removed from the background

Figure 5.4: Comparison between an original reference image and its version without the unwanted background elements

Removing parts of the image's background would not produce features anymore, leading to a more consistent feature set for each image. Since ARFoundation does not provide a way to see the feature points extracted from an image and both Vuforia and ARFoundation use similar methods for feature extraction, Vuforia's image viewer was used to visualize the feature sets extracted from reference pictures.

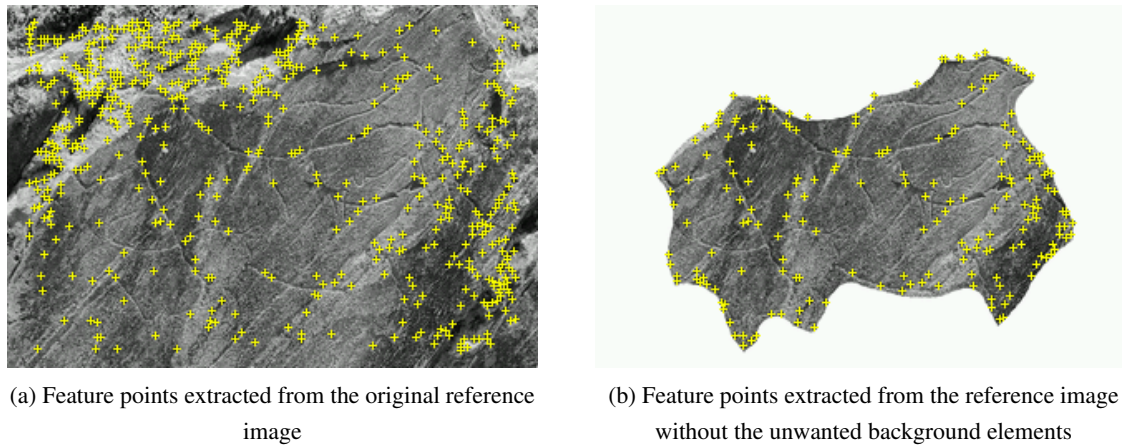


Figure 5.5: Comparison between the feature set extracted from an original reference image and its version without the unwanted background elements

As shown in Figure 5.5, the original reference image contains feature points that are not directly related to the rock itself but to the vegetation elements present in the top-left corner of the picture. These features introduce more complexity to the image detection process and make the reference image harder to detect since the camera must also capture the vegetation elements for a match to be produced. Moreover, removing excessive elements from the original reference images generally improved their quality scores, measured with the *arcoring* tool, as shown in Table 5.5.

	Quality Score
Original Image	20
Image without background	45

Table 5.5: Quality scores obtained by each version of the reference image

Table 5.6 shows the changes observed in each performance metric measured.

Metric	Result
1.a	Improved
1.b	Improved
2.a	No change
2.b	No change
2.c	No change

Table 5.6: Performance metrics results for the approach of using multiple reference images for each engraving/rock

After testing this approach and comparing it to the baseline version of the application, users reported improvements in the initial detection of the target rock, feeling that this step was significantly faster in the prototype version that implemented this approach. Moreover, more views of the

target rock were able to be detected, therefore contributing to higher robustness. Regarding step 2, no notable changes were noticed by the users for metrics 2.b and 2.c. Metric 2.a was measured in both approaches and there were no changes in its value, as shown in Table 5.7.

Approach	FPS
Baseline	30
Removed unwanted background elements	30

Table 5.7: Number of frames processed per second by the application using each tested approach

This alternative was expected to significantly impact the first step of the image detection and tracking process, which was confirmed after its implementation and testing. Although no notable improvements were made regarding step 2, all the metrics for this step remained unchanged. Therefore, this technique was implemented in the final version of the application.

5.2.3 Preprocessing the camera input

After testing the approach where multiple reference images were used for each object to be detected, described in section 5.2.1, and observing that changes in the brightness and contrast values of the reference images produced better results than just using the original picture, the idea of applying the same concept to the camera input arose. Contrast and brightness can normally be adjusted in the camera settings of mobile devices. However, when executing an AR application based on ARCore, like the developed solution, the framework takes full control of the camera and only exposes some parameters, like the resolution or the exposure, to the developer, not allowing access to the others. Since the contrast and brightness parameters were not accessible at runtime, this idea had to be implemented differently. The alternative implementation consists in taking each frame captured by the camera as an image and applying the changes to it before passing the captured picture to the tracking module, as shown in Figure 5.6.

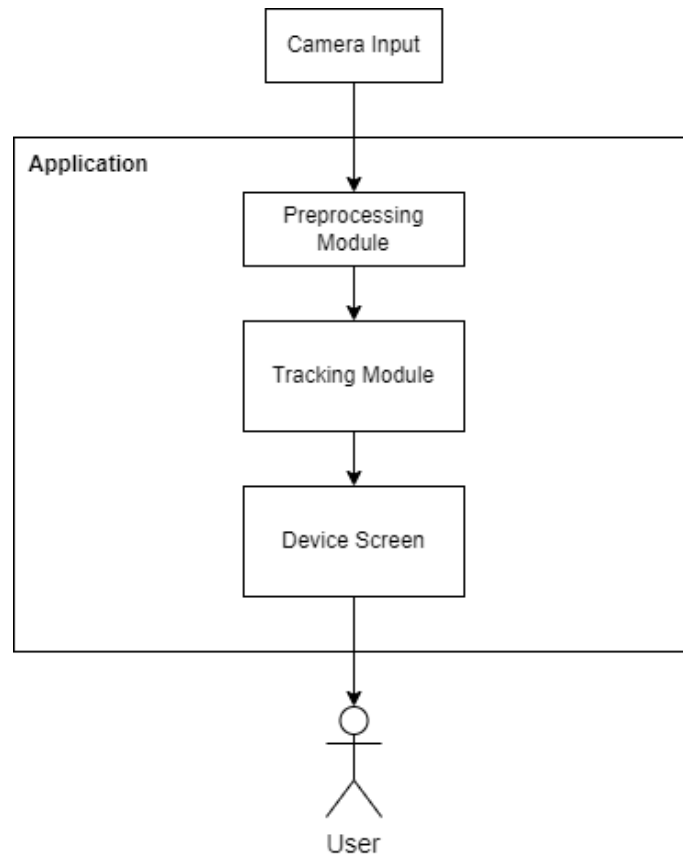


Figure 5.6: Diagram that shows the flow of preprocessing the camera input before using it for tracking

This process involved keeping a copy of the captured frame so that it could be altered for tracking purposes. The changes are not visible to the user since the original captured frame is the one rendered on the device's screen, but the tracking module uses the altered version of the frame to extract features. This approach introduces a higher computational cost to the application, which impacts its performance. As shown in Table 5.8, the average number of frames processed by the application per second was lower when the camera input was being preprocessed. Although it did not make the application unusable, this impact on the performance was noticeable to the user and negatively impacted the experience.

Approach	FPS
Baseline	30
Preprocessed camera input	26

Table 5.8: Average frames per second achieved by the application using each approach, measured on a Xiaomi Redmi 10T Lite phone

The results achieved by this approach regarding the efficiency of detecting reference images and tracking them were inconclusive. While increasing the contrast of the captured camera frames

did improve the initial detection of reference images, their tracking became unstable, with imprecise pose estimation calculations and image tracking lost when the camera suffered sudden movements. The test results regarding contrast changes in the camera input, displayed in Table 5.9, show that step 1 was slightly affected by these changes but step 2 was severely worsened. Metric 2.a displays a decrease in performance, as expected, generated by the additional computational operations required to implement this approach. Furthermore, the accuracy of the estimated position of the engravings (Metric 2.b) was also reported to be worse than the baseline version, with occasional drifts in the position's estimate. Metric 2.c reveals the instability that this approach introduced to the tracking of the reference image, given that it would occasionally get lost without the image leaving the camera view.

Metric	Result
1.a	Slightly improved
1.b	No change
2.a	Worsened
2.b	Worsened
2.c	Worsened

Table 5.9: Performance metrics results for the approach of increasing the contrast of the frames captured by the camera

Brightness changes did not improve image detection and tracking, even in darker scenarios, contrary to what was expected. As shown in Table 5.13, no changes were reported for metrics 1.a, 1.b, 2.b and 2.c. Metric 2.a reflects the decrease in performance introduced by this method.

Metric	Result
1.a	No change
1.b	No change
2.a	Worsened
2.b	No change
2.c	No change

Table 5.10: Performance metrics results for the approach of increasing the brightness of the frames captured by the camera

Overall, this approach did not improve either step of the image detection and tracking process and negatively impacted the application's performance. Therefore, this technique was not included in the final solution.

5.2.4 Using outlines as reference images

The tracking module takes the features extracted from the camera input and compares them to the feature sets in the reference database. Therefore, in order to detect a certain image, the image itself does not need to be used as reference, one only needs to ensure that the reference image contains a subset of the original image's feature set. This is the same concept applied to the approach explained in section 5.2.2, where elements present in the background of reference images are removed. The features extracted from the resulting image are a subset of the original reference picture's feature set, ensuring that the original image is still detected if the altered reference picture is used.

Based on this concept, the idea of using the outlines of each engraving as reference images arose. Figure 5.7 shows an example of an outline extracted from one of the engravings in rock number 6 in the Penascosa rock art site. Since the outlines would have the same shape as the engraving itself, the features they produced should match the ones extracted from the original image.

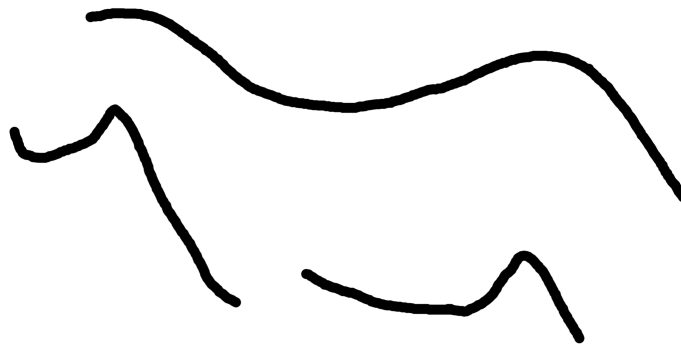


Figure 5.7: Outline extracted from the engraving representing a horse

The outlines of the multiple engravings present in the target rock were extracted and tested. After analyzing the extracted feature points using Vuforia's image viewer, it was noted that the outlines produced a significantly smaller set of features than the original image. Furthermore, since the outlines were extracted manually, their shape was prone to human error and, although very close to the original engraving, was not exactly the same, which meant that the features represented a slightly different shape than the original engraving. Figure 5.8 shows the feature points extracted from the horse engraving outline displayed earlier.

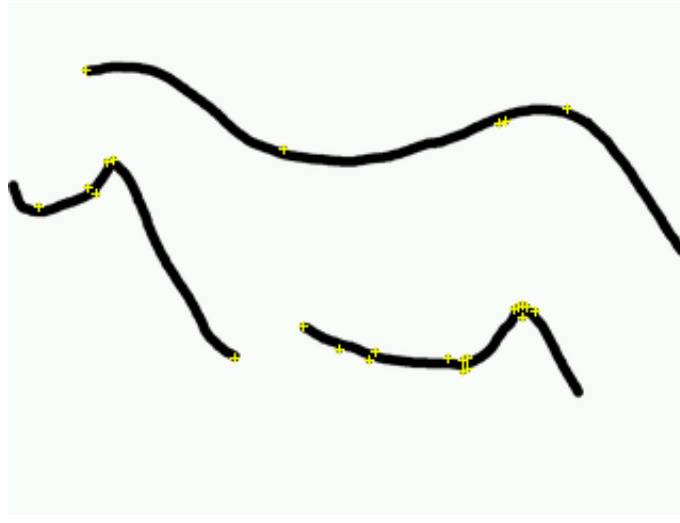


Figure 5.8: Features extracted from the horse engraving outline

From this picture, it is noticeable that the number of feature points extracted is too small to represent the shape of the engraving correctly.

Apart from this, when the engravings were small, the outlines tended to have thicker lines, interfering with the feature extraction. As seen in Figure 5.9, both sides of the lines produce features, which did not happen in the original engraving images. This introduces new features into the feature set that are not contained in the original engraving.

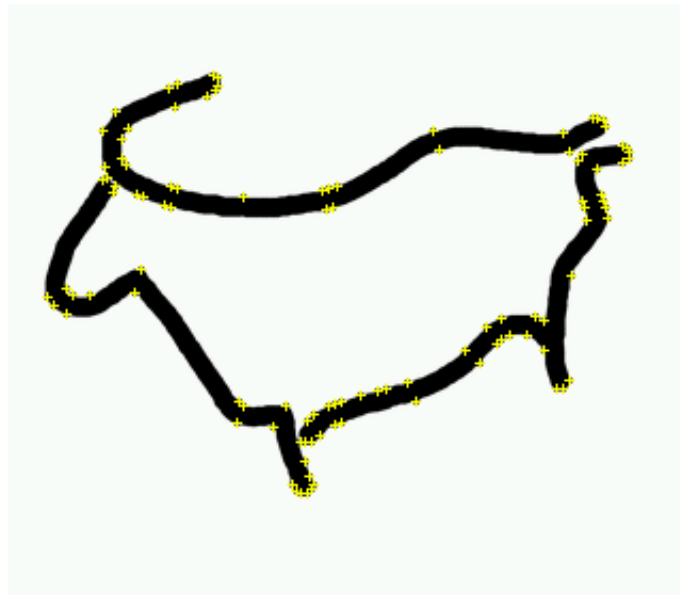


Figure 5.9: Features extracted from one of the goat engraving outlines

Furthermore, the *arcoreimg* tool, which evaluates the quality of a reference image was not able to provide a quality score to any of the outlines used due to the outlines not having a sufficient number of keypoints to be used as features, as shown in Table 5.11.

	Quality Score
Horse outline	Not enough keypoints
Goat 1 outline	Not enough keypoints
Goat 2 outline	0

Table 5.11: Quality score obtained by each outline, calculated with the *arcoring* tool

Despite these preliminary problems, this approach was implemented and tested. Multiple outlines were used for the same engraving to minimize the human error introduced when manually extracting them. This approach was tested in six different setups for each engraving. Two different outlines were extracted from the same engraving and each of them was tested with three different pictures of the rock's frontal view. The goal was to assess if the tracking module would detect the original engraving using just the outlines as a reference. Table 5.12 shows, for each engraving present in the target rock, the number of times they were detected using this approach.

	Number of Matches (using 6 different setups)
Horse engraving	0
Goat engraving 1	0
Goat engraving 2	0

Table 5.12: Number of matches achieved by using each outline as reference image

This approach did not work as intended initially and its implementation confirmed the results obtained in the preliminary tests, presenting a decrease in two of the five measured performance metrics, as shown in Table 5.13. The other three metrics were not measured given that the application was unable to reach a point where the operations related to each of them could be performed. Both metrics related to the first step were severely worsened, with users reporting that the application could not detect the target rock in any of the views presented. Since the application was unable to reach step two in any of the tests, it was impossible to measure metrics 2.b and 2.c. Metric 2.a was measured during the first step and did not present any changes from the baseline version, with the number of frames processed by the application per second remaining stable.

Metric	Result
1.a	Severely worsened
1.b	Severely worsened
2.a	No change
2.b	Not measured
2.c	Not measured

Table 5.13: Performance metrics results for the approach of increasing the brightness of the frames captured by the camera

As seen in Table 5.12, the tracking module was not able to detect the original engraving using the extracted outlines for any of the engravings, which led to a significant decrease in performance overall. Due to the poor results obtained by this approach, it was not included in the final solution.

5.2.5 Phone flashlight

Early in the development stage, the focus was mainly on improving image detection and tracking efficiency across different lighting settings. The idea of using the mobile device's flashlight to boost the luminance in low-light conditions arose because darker scenarios generated worse outcomes than brighter ones did. Before trying to implement this approach, its concept was tested using an external flashlight instead of the one from the device. These tests were performed only with the printed versions of the target rock's pictures since displaying the images on a computer was unviable due to the screen reflecting the flashlight's brightness. These preliminary tests achieved positive results, with most users reporting an improvement in their experience. As shown in Table 5.14, improvements were registered only in metric 1.a, which regards the time for the initial detection of a reference image. The registered improvements were only noted in dark scenarios, where image detection proved challenging.

Metric	Result
1.a	Improved in dark scenarios
1.b	No change
2.a	No change
2.b	No change
2.c	No change

Table 5.14: Performance metrics results for the approach of using an external flashlight to increase the luminance of the environment

Since the preliminary results were positive, it was decided that this approach would be implemented. However, at the time of development, ARCore's session takes full control of the device's camera and does not allow the developer to control the flashlight. Therefore, this feature could not be implemented and inserted into the application, despite its potential for improving step 1.

Chapter 6

Usability Evaluation

This chapter presents and discusses the results of the usability evaluation tests. The test sessions' setup is presented in section 6.1, followed by a description of their structure, in section 6.2. The tasks used in each session are detailed in section 6.3 and the results are presented in section 6.4.

6.1 Setup

All the tests carried out in a simulated environment were conducted in person. The following hardware and materials were needed for each study:

- One mobile phone to execute the application.
- One laptop to show pictures of rock number six of the Penascosa rock art site.
- One laptop for users to answer the evaluation questionnaires.
- Printed pictures of the target rock.

The application was executed on the same mobile device for all tests to maintain the highest possible level of consistency across them. The device used was a Xiaomi Mi 10T Lite.

6.2 Session Structure

Although not all tests were conducted in the same location, all were carried out in comfortable environments for the users.

Firstly, users were briefed about the structure of the test session they were participating in. It was explained to all participants that they could freely ask questions about the experience and the questionnaires. It was also explained that they could leave the session at any point if they felt uncomfortable.

After the introductory briefing, users were asked to answer a demographic questionnaire and were asked about their experience with mobile AR applications and with AR itself. Users who demonstrated comfort using AR applications and had previous experience with the technology

were presented with the application's context and the role AR plays in its experience. A more in-depth explanation regarding AR was presented to those who reported not having experience with such applications. The Penascosa rock art site was also presented to all participants through a verbal description and pictures to give the users a better idea of the real-world conditions in which the solution should be used.

The test experience started after introducing the application. In this phase, participants were presented with different task scenarios and had to complete certain tasks using the provided mobile device and the pictures of Penascosa's rock number six. After each step, users were asked to answer questions about the completed tasks.

The test experience ended when the users finished the last task and answered its respective questions. After this, a System Usability Scale (SUS) questionnaire was given to the participants to fill and, finally, they were asked for general feedback regarding the application as a whole. Users were also able to give suggestions for changes or improvements that could be implemented in the solution.

The following list summarizes the structure of each test session:

1. Introductory briefing
2. Filling in the demographic questionnaire
3. AR briefing
4. Introduction to the application and its context
5. Application testing based on specific tasks
6. Filling in the task-related forms
7. Filling in the SUS questionnaire
8. Open feedback and suggestions

6.3 Tasks

In order to test the usability of the application, two tasks were created for users to follow during the test sessions. A task scenario was created to give the users a better sense of the real-life scenario in which the application would be used. The created tasks share the task scenario and two questions but differ in the remaining ones. This section will describe the task scenario, as well as the structure and goals of each task.

6.3.1 Task Scenario

To provide users with a better context of the real scenario where the application would be used, they were given a task scenario before performing the tasks. Since both tasks share the same scenario, this was only described to the user once, before starting the first task.

The scenario was the following:

"The user decided to visit the Côa Museum and the Penascosa rock art site. After completing the museum visit, the participant was inserted into a group of six people who were taken to the Penascosa site by a guide. Upon reaching the location of the engravings, the guide introduced the rocks and some of the engravings, explaining their story and context, and then gave the visitors some time to explore the engravings on their own and provided them with mobile devices containing the AR application."

6.3.2 Task A - Visualizing a 2D animation

This task requires the user to perform three steps: **detect rock engravings (A.a)**, **select one of the engravings (A.b)** and **visualize a 2D animation of the selected engraving (A.c)**.

Step A.a aims to evaluate the image detection and tracking module, determining its ease of use. Step A.b has the goal of evaluating the usability of the Engraving Detection Page UI that allows specific engravings to be selected. Finally, step A.c aims to assess the performance of the 2D animation player and understand if displaying the animations on the scene itself is well accepted by the users.

This task has 6 associated questions, which users were asked to answer. Questions are answered using a Likert Scale, with 5 possible answers:

1. *Strongly Disagree*
2. *Disagree*
3. *Neutral*
4. *Agree*
5. *Strongly Agree*

The questions used for this task are the following:

- **Question 1 (QA1)** (related to step A.a) - *"I was able to detect rock engravings using the application easily"*
- **Question 2 (QA2)** (related to step A.a) - *"The engraving detection process was simple and intuitive"*
- **Question 3 (QA3)** (related to step A.b) - *"I was able to associate each button on the bottom panel with its respective engraving"*
- **Question 4 (QA4)** (related to step A.b) - *"Selecting a specific engraving was simple and intuitive"*
- **Question 5 (QA5)** (related to step A.c) - *"The animation button was easy to locate"*

- **Question 6 (QA6)** (related to step A.c) - *"I was able to visualize the 2D animation clearly"*

Each step has two related questions. Table 6.1 shows which questions are associated with each step.

Step	Questions
A.a	QA1, QA2
A.b	QA3, QA4
A.c	QA5, QA6

Table 6.1: Task A steps and questions

6.3.3 Task B - Visualizing and controlling a 3D model

This task requires the user to perform the same two initial steps as task A. As these steps were already evaluated in that task, they were not included in this one. Therefore, to complete this task, the user has to perform the following steps, starting after engravings are detected and selected: **enter the 3D Visualization Page (B.a)**, **detect horizontal planes in the real-world scene (B.b)**, **place the 3D model on the scene (B.c)** and **control the model's rotation and scale (B.d)**.

Step B.a is designed to evaluate the intuitiveness of the navigation needed to access the 3D Visualization Page. The plane detection module is evaluated in step B.b and the simplicity of placing a model on the scene in step B.c. Finally, step B.d aims to assess the ease of use of the model control UI, as well as its intuitiveness.

The questions used for this task are the following:

- **Question 1 (QB1)** (related to step B.a) - *"The 3D visualization button was easy to locate"*
- **Question 2 (QB2)** (related to step B.b) - *"The plane detection hint helped me understand how to detect planes"*
- **Question 3 (QB3)** (related to step B.b) - *"I was able to easily detect horizontal planes"*
- **Question 4 (QB4)** (related to step B.b) - *"Detected planes were easy to identify"*
- **Question 5 (QB5)** (related to step B.c) - *"Placing the 3D model on the scene was simple"*
- **Question 6 (QB6)** (related to step B.d) - *"Rotating the model was easy and intuitive"*
- **Question 7 (QB7)** (related to step B.d) - *"Controlling the model's scale was easy and intuitive"*

Table 6.2 shows which questions are related to each step.

Step	Questions
B.a	QB1
B.b	QB2, QB3, QB4
B.c	QB5
B.d	QB6, QB7

Table 6.2: Task B steps and questions

6.3.4 General Questions

After completing both tasks, participants were asked to fill out a questionnaire containing five Likert Scale questions, with five options, ranging from *Strongly Disagree* to *Strongly Agree*, regarding the application overall. The questions were the following:

- **Question 1 (QC1)** - *"Navigating across the application was intuitive and simple"*
- **Question 2 (QC2)** - *"The buttons were easy to locate"*
- **Question 3 (QC3)** - *"The loading speed of the different pages was good enough to affect the experience of using the app negatively"*
- **Question 4 (QC4)** - *"This application improved my ability to identify and interpret rock engravings"*
- **Question 5 (QC5)** - *"I would like this application to be included in rock art visits"*

The first question (QC1) aims to evaluate the navigation across multiple application pages. Question QC2 has the objective of understanding whether the UI buttons were easily identifiable and located. The third sentence regards the loading speed of the application pages and its impact on the user experience. Finally, questions QC4 and QC5 have the goal of assessing the helpfulness of the application in rock art visits.

6.4 Results

This section presents the results obtained from the test sessions conducted.

6.4.1 Participants Population

Most participants were male (66.6%), while only 33.3% were female, as shown in Table 6.3. Regarding their ages, most had an age of less than 30 years old and only three users were above 40 years old, as demonstrated in Table 6.4. Table 6.5 shows that participants' ages ranged from 19 to 52, with a mean of 28.11 and a median of 23.5 years old. Regarding the users' past experience with AR, especially mobile AR applications, 88.8% reported having used AR applications in the

Gender	Number of Participants
Male	12
Female	6
Other	0

Table 6.3: Number of participants of each gender

past, while 11.1% (2 participants) mentioned not having experience with AR applications, as seen in Table 6.6.

Age	Number of Participants
18-22	6
23-27	7
28-32	2
33-37	0
38-42	0
43-47	1
48-52	2

Table 6.4: Number of participants of each age range

Participants' Age	
Minimum	19
Maximum	52
Mean	28.11
Median	23.5

Table 6.5: Statistics regarding participants' age

Past Experience with AR	Number of Participants
Yes	16
No	2

Table 6.6: Number of participants of each age range

6.4.2 Task A - Visualizing a 2D animation

Users were given pictures of Penascosa's rock number six engravings in both digital and printed formats and were allowed to choose which picture and format to use. In order to complete this task, users were required to visualize a 2D animation on top of one of the engravings. Since the

horse engraving was the only one with animation, this was explained to the participants before starting the task.

The questions users had to answer for this task are the following:

- **Question 1 (QA1)** (related to step A.a) - *"I was able to detect rock engravings using the application easily"*
- **Question 2 (QA2)** (related to step A.a) - *"The engraving detection process was simple and intuitive"*
- **Question 3 (QA3)** (related to step A.b) - *"I was able to associate each button on the bottom panel with its respective engraving"*
- **Question 4 (QA4)** (related to step A.b) - *"Selecting a specific engraving was simple and intuitive"*
- **Question 5 (QA5)** (related to step A.c) - *"The animation button was easy to locate"*
- **Question 6 (QA6)** (related to step A.c) - *"I was able to visualize the 2D animation clearly"*

All the questions were based on the Likert Scale and had 5 possible answers, each associated with a numerical value:

1. *Strongly Disagree*
2. *Disagree*
3. *Neutral*
4. *Agree*
5. *Strongly Agree*

The questions were all answered after completing the task.

Results

Figures 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6 show the answers to each question for this task aggregated in bar charts.

Regarding question QA1, 77.7% of users answered either *Agree* or *Strongly Agree*, demonstrating that users did not find the process of detecting engravings (step A.a) difficult. Four participants were neutral regarding this question, which shows that this flow can still be improved. Question QA2 is similar to QA1 by also targeting step A.a, but focuses more on its simplicity and intuitiveness. The answers to this question were identical to QA1's, although one participant disagreed that the engraving detection process was simple and intuitive.

Questions QA3 and QA4 are related to the engraving selection operation (step A.b). 61.1% of users were clearly able to associate each button on the UI with its respective engraving. No

participant disagreed with QA3, showing that the button design used is adequate for their function. All the responses to QA4 were positive, with 72.2% of users strongly agreeing with the affirmation and 27.8% agreeing. There were no negative replies to this question. These results indicate that step A.b was appreciated by the participants.

Regarding step A.c, users reported the animation button was easy to identify and locate, with 77.8% of participants strongly agreeing with that affirmation and without any negative or neutral responses to question QA5. Regarding question QA6, related to the clearness of the 2D animation, responses were not as clear as in the last question. The majority of users (55.6%) answered positively, reporting they were able to clearly see the 2D animation being played on the scene. Five participants were neutral about this matter, while three expressed negative opinions. Two users (11.1%) strongly disagreed with question QA6.

Analysis

Overall, the feedback obtained from task A's questions was positive, with users reporting being satisfied with the flow of most steps.

Regarding step A.a, the engraving detection step, most users found it easy and simple, although there were few *Strongly Agree* answers. Some participants reported not finding this process very intuitive and needed help in order to complete this step. During the tests, help was only provided to users when they were stuck in one step. In an actual visit to the rock art sites, the guide can provide users with the same level of help provided in the test sessions, and that is why participants were aided. One way to make this step more intuitive and simple would be to add a helping screen to the Engraving Detection Page. This screen could be displayed in different ways, although there are two main ones: 1) as a temporary hint and 2) as a helping panel. The first option would consist of a piece of text containing instructions on how to perform step A.a that would be temporarily displayed on the screen after some time without detecting any engravings. The second way consists in having a help button in the Engraving Detection Page UI that, when clicked, displays a panel with instructions on top of the page. The panel would also contain a button to close it. Regardless of how they are displayed on screen, instructions and tips would be helpful to users and would improve their experience.

Step A.b achieved the best results overall, with both associated questions having a mean score of above 4.50. Question QA3 shows that the selection UI containing a panel on the bottom of the screen and a button for each engraving was adequate for this use case. 61.1% of participants strongly agreed with the affirmation of QA3 regarding the association between each engraving and its respective button. The only complaint mentioned by some users is that the buttons do not change when selected and, therefore, it is not possible to identify which engraving is currently selected just by looking at the buttons. This is not a problem for most rocks, but could pose an obstacle in cases where it is more difficult for users to constantly keep the rock in the screen. In such cases, it would be helpful to have some distinction between the button of the selected engraving and the others. This could be done by applying some transparency to the buttons of

the engravings that are not selected and keeping the selected one fully opaque, highlighting the pressed button and giving users a way to know the selected engraving at all times.

Regarding the animation step (step A.c), users mostly found the animation button easy to locate and identify. However, some participants mentioned the button text not being clear enough and suggested changing the current text, "PLAY", to "ANIMATION" or "PLAY ANIMATION". Regarding the animation itself, being displayed on the scene proved to be confusing for some users that reported not being able to clearly visualize it. The on-scene display means that the animation's pose is the same as its respective engraving and, therefore, it is played at the same position as the engraving's virtual outline. Since outlines present some small oscillations in their position when the user moves the camera due to the tracking module's position estimation calculations, users often thought these oscillations were part of the 2D animation. While playing the animation, these oscillations can also occur, meaning that the virtual outline might slightly change its position, which can affect the clearness of the animation itself. Improving this aspect means making it more clear to the users that the animation is being played so that they can fully focus their attention on it. Furthermore, highlighting the part of the screen where the animation is occurring could also help users distinguish between the animation frames changing and the position oscillations. This could be achieved by adding a border around the virtual outline during the animation's duration. This border would help users identify its location, allowing them to adjust their device's position in the real world to better view the animation. Although adding an element wrapping the virtual outline would be helpful in this aspect, it would also increase the noise in the scene. Therefore, testing this addition with users before adding it to the application would be important.

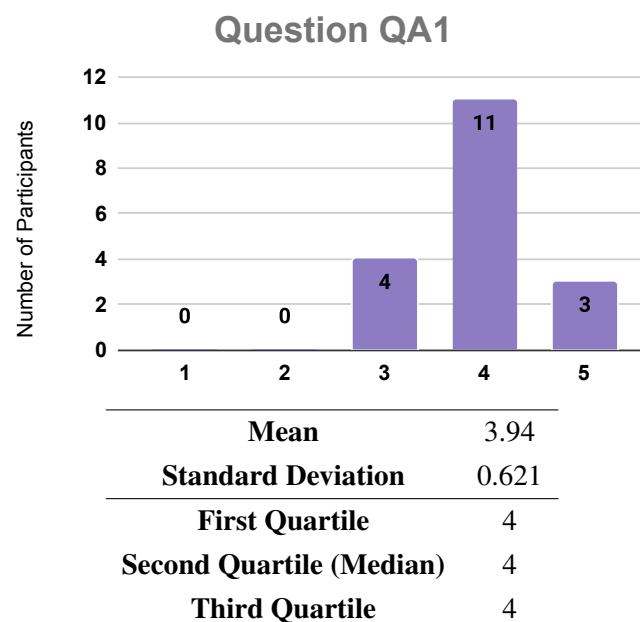


Figure 6.1: Question QA1 Results

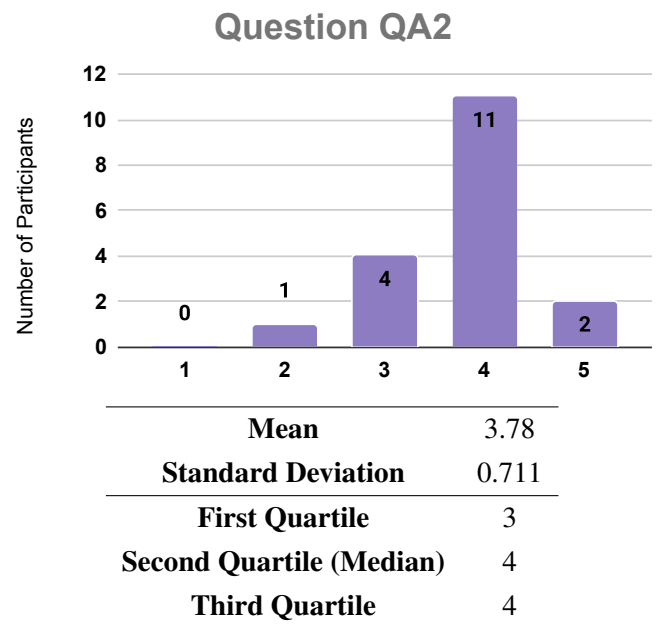


Figure 6.2: Question QA2 Results

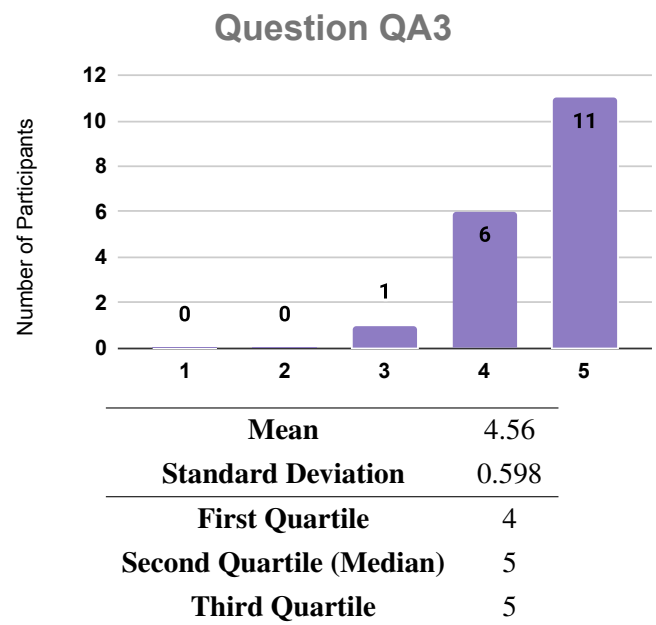


Figure 6.3: Question QA3 Results

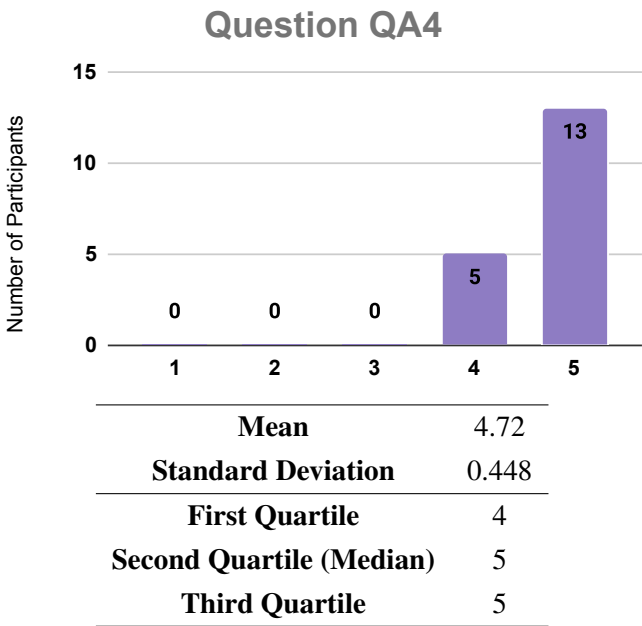


Figure 6.4: Question QA4 Results

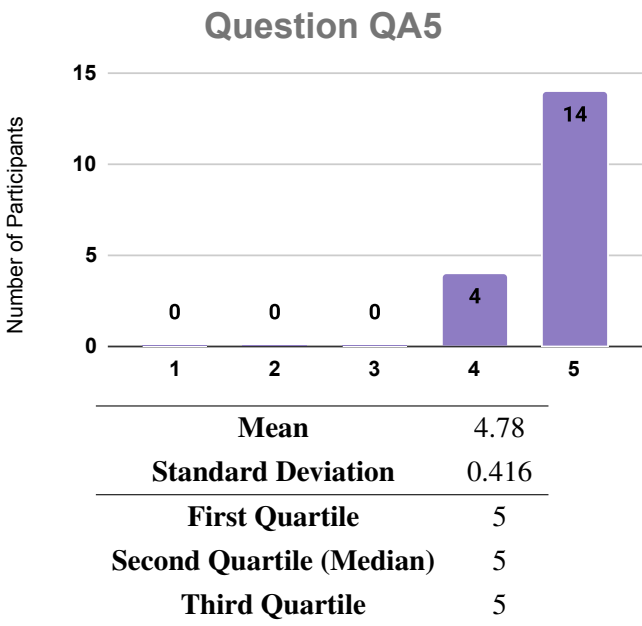


Figure 6.5: Question QA5 Results

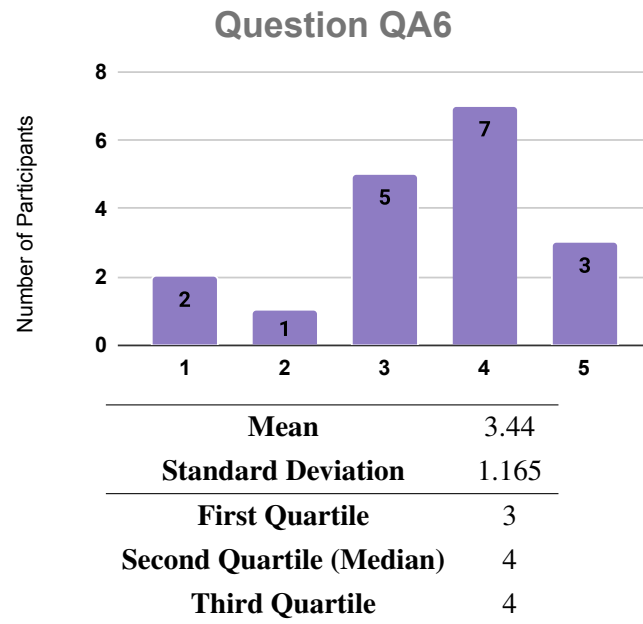


Figure 6.6: Question QA6 Results

6.4.3 Task B - Visualizing and controlling a 3D model

After finishing Task A, the application is on the Engraving Detection Page, engravings have already been detected and one of the engravings has been selected by the user. Task B starts at this point, after an engraving has been selected, which is where Task A finished. To complete Task B, users are required to visualize a 3D model of the selected engraving and control its size and orientation.

The questions related to this task are the following:

- **Question 1 (QB1)** (related to step B.a) - *"The 3D visualization button was easy to locate"*
- **Question 2 (QB2)** (related to step B.b) - *"The plane detection hint helped me understand how to detect planes"*
- **Question 3 (QB3)** (related to step B.b) - *"I was able to easily detect horizontal planes"*
- **Question 4 (QB4)** (related to step B.b) - *"Detected planes were easy to identify"*
- **Question 5 (QB5)** (related to step B.c) - *"Placing the 3D model on the scene was simple"*
- **Question 6 (QB6)** (related to step B.d) - *"Rotating the model was easy and intuitive"*
- **Question 7 (QB7)** (related to step B.d) - *"Controlling the model's scale was easy and intuitive"*

Like Task A, all questions are Likert-Scale based and have five possible answers, each associated with a numerical value, as described in section 6.4.2. The questions were answered after

completing the task.

Results

Figures 6.7, 6.8, 6.9, 6.10, 6.11, 6.12 and 6.13 show the answers to each question for this task aggregated in bar charts.

Question QB1 regarded the button that initializes the 3D Visualization Page. All answers to this question were positive, with 16.6% of participants agreeing that the button was easy to locate and 83.3% strongly agreeing with that affirmation. This demonstrates that accessing the 3D Visualization Page is perceived as a simple and intuitive process by the users.

Questions QB2, QB3 and QB4 are related to the second step of this task (step B.b), the detection of horizontal planes in the real-world scene. 83.3% of users demonstrated finding the plane detection hint to be helpful in the plane detection process, while three users were neutral about this matter. Regarding the plane detection itself, targeted by question QB3, most users (61.1%) agreed that this process was easy and three users strongly agreed with this sentence. One participant disagreed with the affirmation and three users remained neutral. Regarding QB4, all the answers expressed positive feedback, with 72.2% of users strongly agreeing that the planes were easy to identify and the rest agreeing with that statement.

Question QB5 was the only one related to step B.c, the placement of the model on the scene. 77.8% of users were positive about this step. One user disagreed that placing the model on the scene was simple and three participants were neutral about this topic.

Finally, questions QB6 and QB7 targeted step B.d: controlling the 3D model. Regarding the rotation controls, explored in question QB6, the results were very positive. Out of the 18 participants, 17 strongly agreed that rotating the model was easy and intuitive while the remaining user also agreed with the statement. The scale controls (question QB7) achieved slightly worse results, but all users still agreed (16.7%) or strongly agreed (83.3%) that these controls were easy to use.

Analysis

Question QB1, which regards the process of accessing the 3D Visualization Page, achieved great results. Its mean score of 4.83 reflects that users did not have any trouble locating the button that navigates to that page and understanding its function. Some users reported that the combination of the icon and text used in the button provided a very clear idea of what that UI element does.

Regarding step B.b, users were mostly satisfied with the hint shown before any horizontal plane was detected. Participants reported that this element provides helpful instructions on how to detect planes, although some mentioned the hint not being clearly visible in some cases due to its color. Since the element is colored white, it can become fully or partially omitted if the camera captures real-world elements with bright colors. There are two main ways to solve this issue that might be worth exploring in the future. Firstly, the hint's color could be changed depending on

the overall brightness of the environment captured by the camera. Using Arcore's light estimation features, it is possible to estimate the brightness level of the scene, which could then be used to adjust the color of the hint. In darker environments it would remain white and in brighter ones its color could be changed to black. The second method would be to add a background to the hint element. This would ensure that the hint is always visible, but it would occupy a larger portion of the screen. Regarding the detection of horizontal planes, most users were able to easily perform this step. However, some participants demonstrated difficulties understanding what they were supposed to do, even with the instructions provided by the hint. It could be worth exploring the addition of more UI elements that provide clearer instructions on how to perform this operation. When it comes to the planes themselves, after the application detects them as horizontal surfaces, an orange polygon appears on the scene denoting that the surface where it is placed has been detected. All users were able to understand this concept and could easily point and identify the detected surfaces. Some participants suggested changing the color of the polygons. Since this could seriously affect the view of the real-world scene, this approach would have to be tested with different colors and opacity values before being implemented.

Question QB5 targets the placement of the 3D model on the scene (step B.c). Users were generally satisfied with the actions needed for this step, but most mentioned the steps following the detection of horizontal surfaces were unclear. Although for some participants tapping one of the detected surfaces to place the 3D model on the scene was intuitive, there were others who got stuck in this step. The lack of instructions regarding this phase seems to be the cause of the problems encountered by the participants and, therefore, this point should be improved. UI hints could be displayed during this step, as in step B.b, explaining to users how to place the 3D model on the scene. Moreover, an instructions panel could also be helpful to mitigate the issues participants faced during this phase.

Questions QB6 and QB7 focused on step B.d, with the goal of assessing the usability of the 3D model controls. The application allows users to control two parameters of the model after it is placed on the scene: orientation and size. These questions achieved high mean scores, reflecting the overall satisfaction with the 3D model controls. Regarding the orientation controls (question QB6), users reported finding them intuitive and easy to use. All users were able to understand the function of the rotation slider and all expressed the icon placed inside the slider's panel was useful for that. The size controls (question QB7) achieved similar results, with all users reporting positive feedback, and being able to clearly identify the UI element relative to it. Some participants suggested allowing the size to be controlled using finger pinches given that this action is usually related to changing the size of objects in other applications.

Generally, the feedback obtained from Task B's questions was positive.

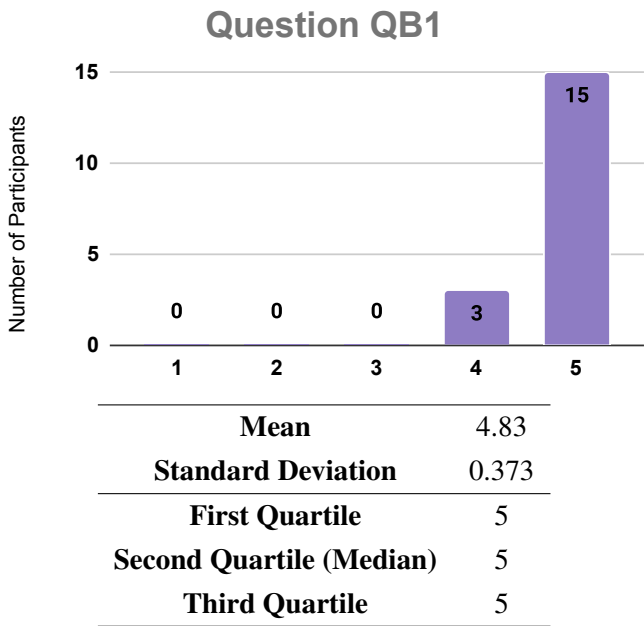


Figure 6.7: Question QB1 Results

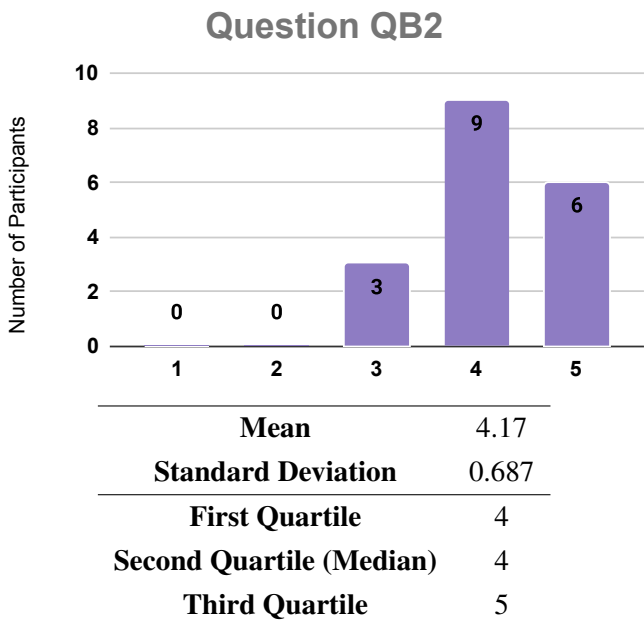


Figure 6.8: Question QB2 Results

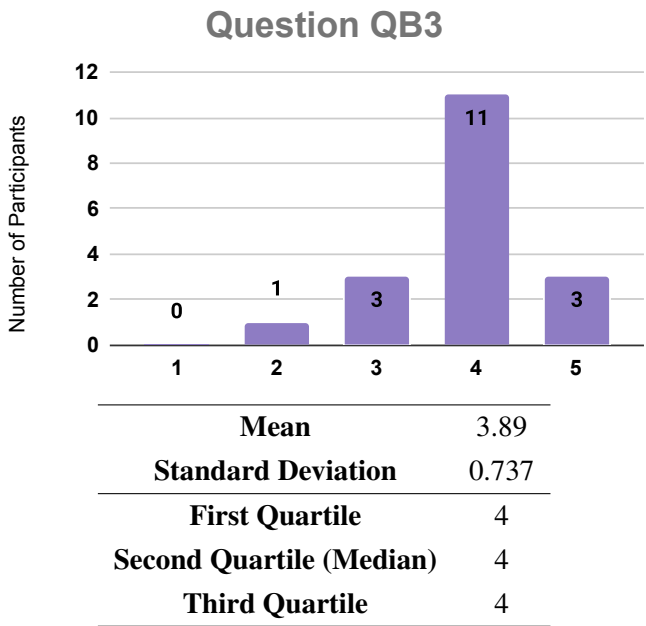


Figure 6.9: Question QB3 Results

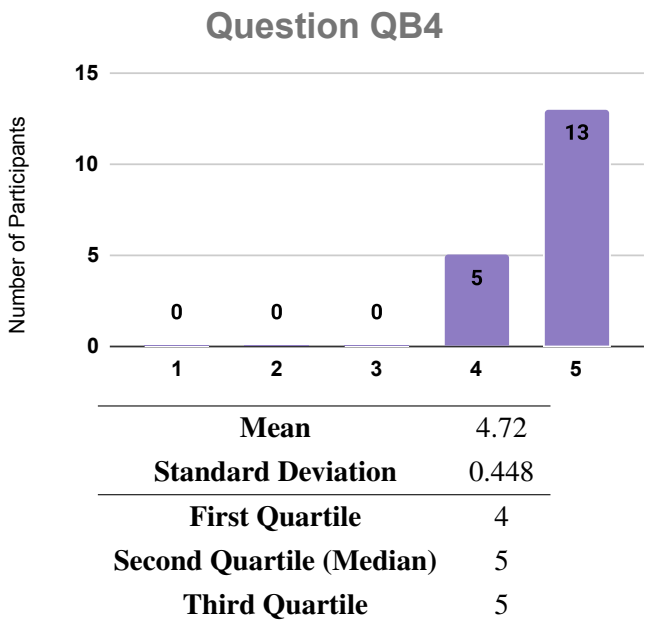


Figure 6.10: Question QB4 Results

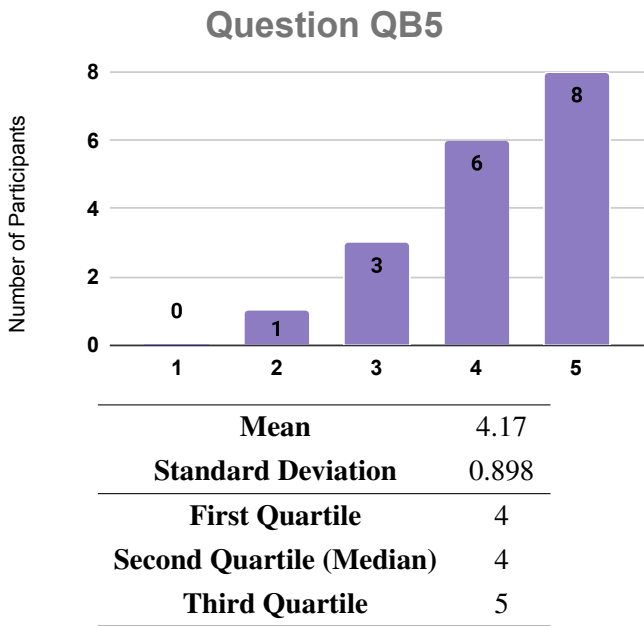


Figure 6.11: Question QB5 Results

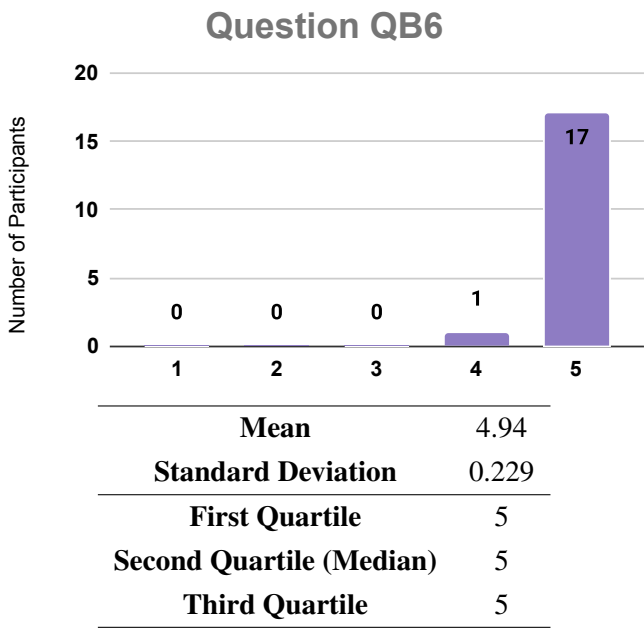


Figure 6.12: Question QB6 Results

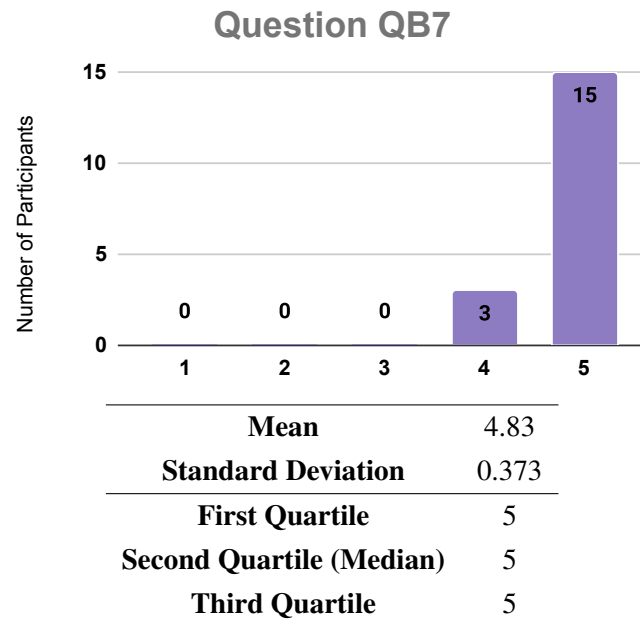


Figure 6.13: Question QB7 Results

6.4.4 General Questions

These questions have the goal of evaluating general aspects of the developed application and, therefore, participants only answered them after completing both tasks and replying to their respective forms.

The general questions are the following:

- **Question 1 (QC1)** - *"Navigating across the application was intuitive and simple"*
- **Question 2 (QC2)** - *"The buttons were easy to locate"*
- **Question 3 (QC3)** - *"The loading speed of the different pages was good enough to affect the experience of using the app negatively"*
- **Question 4 (QC4)** - *"This application improved my ability to identify and interpret rock engravings"*
- **Question 5 (QC5)** - *"I would like this application to be included in rock art visits"*

Like Task A and Task B's questions, these are all based on the Likert-Scale, with five possible answers, ranging from *Strongly Disagree* to *Strongly Agree*. Each answer is associated with a numerical value, as explained earlier in section 6.4.2.

Results

Question QC1 aims to evaluate the navigation across the entire application. 16.7% of users agreed that navigation was simple and intuitive, while the rest (83.3%) strongly agreed with the affirmation. Regarding the buttons (question QC2), 83.3% of users provided positive feedback, either agreeing or strongly agreeing that they were easy to locate and identify. Three users were neutral about this point and there were no negative answers to this question. Question QC3 focuses on the loading speed of the application pages. Regarding this matter, the majority of users (61.1%) agreed that the loading speeds did not impact their experience negatively, while 16.7% strongly agreed with that. Three users were neutral about this aspect and one participant disagreed with QC3's statement, expressing dissatisfaction with the loading speed of the different application pages. QC4 did not register any negative or neutral answers. 72.2% of users strongly agreed that the application helped them identify and interpret rock engravings and 27.8% agreed with that statement, demonstrating that users perceived the application as helpful. Finally, question QC5 aimed to assess the willingness of participants to use this application in rock art visits. Results were mostly positive, with 14 out of 18 participants (77.8%) expressing they would like the application to be included in visits. Three users were neutral about this point and one participant disagreed with QC5's affirmation.

Analysis

Question QC1 demonstrates that navigation across the different application pages is intuitive and simple. Furthermore, QC2 supports this statement, with the majority of users feeling that the UI buttons were easy to locate and understand. Some participants suggested changing the icons or the text of some UI elements, but mostly the navigation feedback was positive. Moreover, users did not demonstrate any difficulties with this aspect and reported feeling confident navigating across the application. Regarding the loading speed of the pages (question QC3), feedback was, once again, positive. Most participants mentioned not feeling like the loading speed of the pages was impacting their experience negatively. Some users reported some discontentment with the loading speed of the Engraving Detection Page when it was opened for the first time. This page takes longer to load than the others because it has to initialize the image detection module. One approach to mitigate this is implementing a loading screen between the Initial Page and the Engraving Detection Page. Although the loading speed would not change, users would, most likely, not feel its impact as strongly. However, this approach would require extensive testing to ensure its acceptance among the final users. Question QC4 demonstrates the application's usefulness in identifying and interpreting rock engravings. Most users reported feeling that the application strongly helped them locate engravings in Penascosa's rock number six and distinguish between them. Overall, the feedback regarding this aspect was extremely positive. Finally, question QC5 aims to determine whether users would like to use the developed application in rock art visits. The majority of participants reported wanting the application to be included in rock art visits. Most participants that expressed this mentioned the visual aid and the liveness of the application's information as the two main reasons for its success. The participants who answered negatively or who

were neutral mentioned feeling like the application had a steep learning curve. Furthermore, there were concerns regarding the repeatability of the solution. This could be solved by regularly updating the contents of the application so that if a user repeats a visit to a certain rock the experience will be completely different.

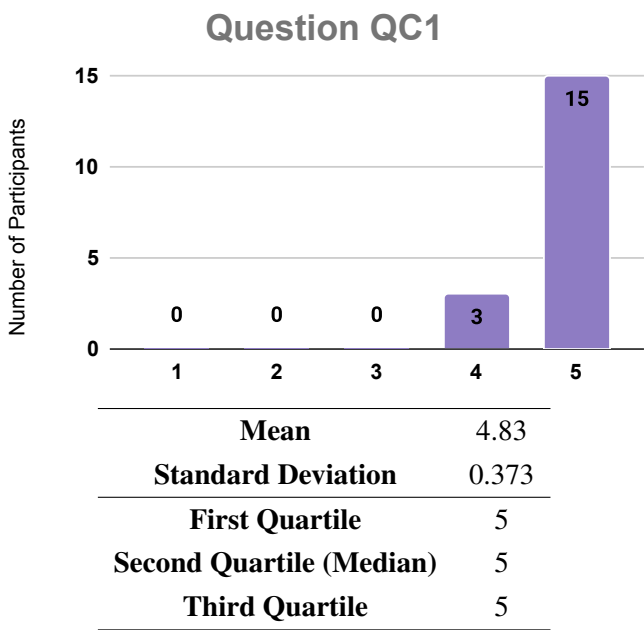


Figure 6.14: Question QC1 Results

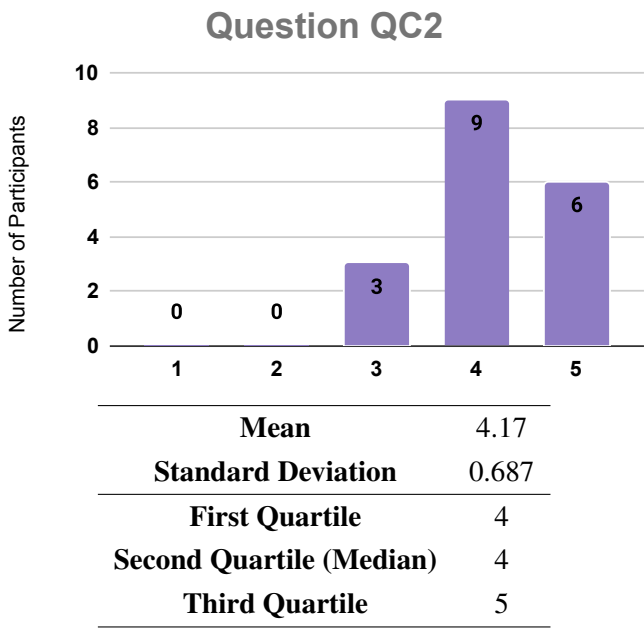


Figure 6.15: Question QC2 Results

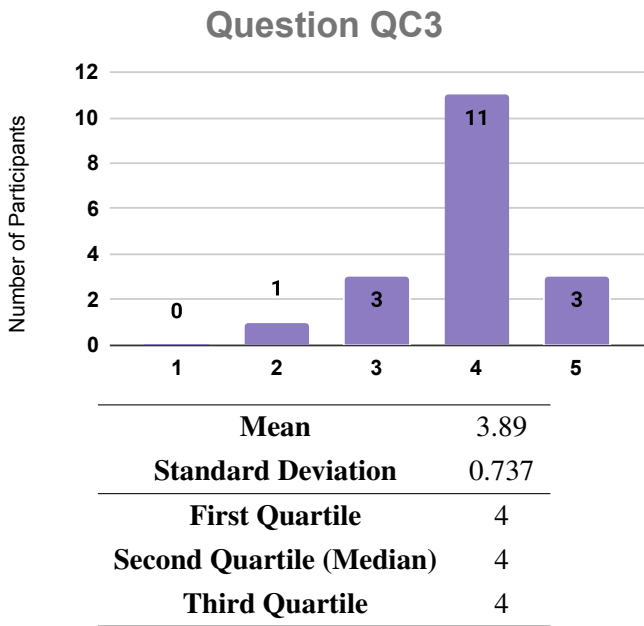


Figure 6.16: Question QC3 Results

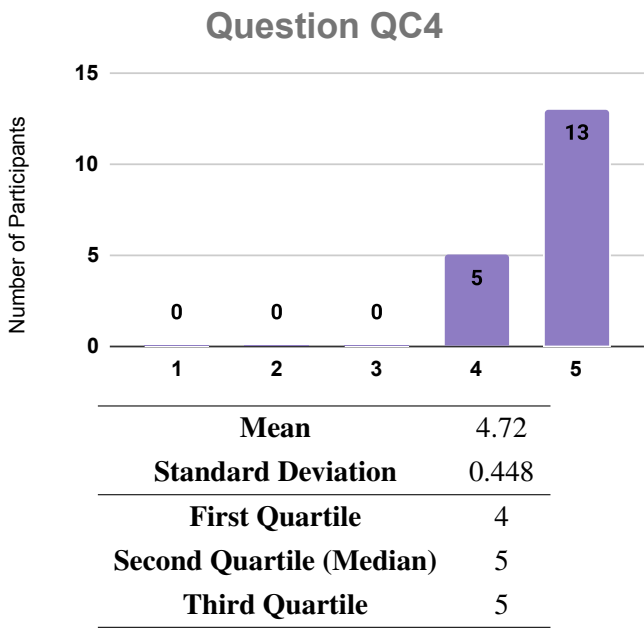


Figure 6.17: Question QC4 Results

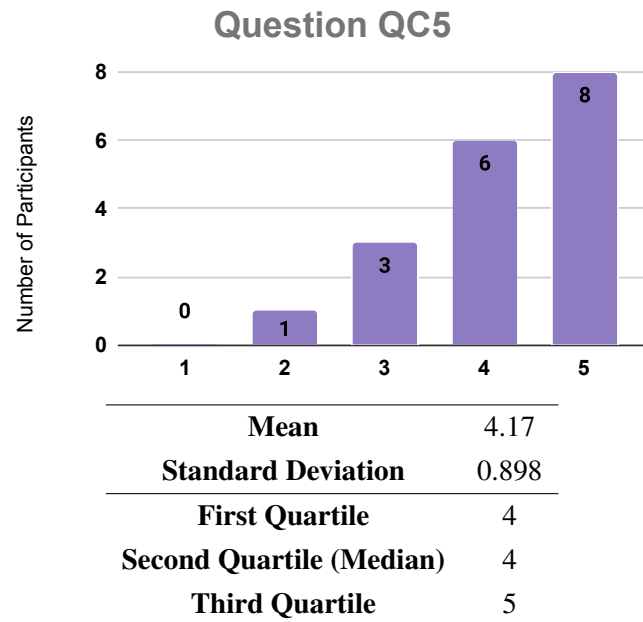


Figure 6.18: Question QC5 Results

6.4.5 System Usability Scale

The System Usability Scale (SUS) questionnaire contains ten Likert-Scale questions. Each question contains a sentence and has five possible answers with numerical values associated, ranging from *Strongly Disagree* (1) to *Strongly Agree* (5). The sentences are the following:

1. *I think that I would like to use this system frequently.*
2. *I found the system unnecessarily complex.*
3. *I thought the system was easy to use.*
4. *I think that I would need the support of a technical person to be able to use this system.*
5. *I found the various functions in this system were well integrated.*
6. *I thought there was too much inconsistency in this system.*
7. *I would imagine that most people would learn to use this system very quickly.*
8. *I found the system very cumbersome to use.*
9. *I felt very confident using the system.*
10. *I needed to learn a lot of things before I could get going with this system.*

Each questionnaire response produces a score. Firstly, the numerical values from each answer (ranging from 1 to 5) have to be scaled in the following way:

- For odd-numbered questions (*score* is the raw numerical value and *scaledScore* the resulting one):

$$scaledScore = score - 1$$

- For even-numbered questions (*score* is the raw numerical value and *scaledScore* the resulting one):

$$scaledScore = 5 - score$$

After this, all scores have values between 0 and 4. The final result is obtained using the following formula, where Q_i is the scaled score of question i):

$$result = (\sum_{i=1}^{10} Q_i) * 2.5$$

The obtained results are in the range of 0 to 100.

The results were interpreted using the Adjective Rating Scale developed by Bangor et al. [5], shown in Figure 6.19, obtained from their article.

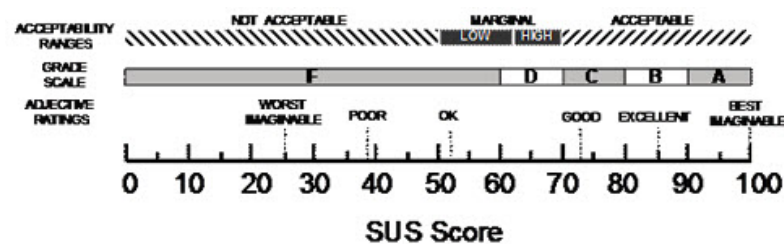


Figure 6.19: Different scales for interpreting a SUS score,

The developed application achieved a mean score of 80.7 from all the answers to the SUS questionnaire. This average score places the solution in the *Acceptable* range according to the acceptability scale, with a *B* grade, according to the grade scale. The adjective rating obtained is *Excellent*. This reflects the general feedback obtained from the test participants, who expressed their satisfaction with the solution.

6.4.5.1 Open Feedback

Users were able to provide open feedback and make suggestions after replying to the SUS form. From all the answers gathered, three points stand out as possible changes to the application in the future:

- **Being able to move the 3D models**

Some users mentioned that being able to see the 3D models move in the scene would be a great addition to the app, increasing the virtual models' feeling of reality. According to the test participants' feedback, this movement should either be controlled by the user or integrated into the model's animation. Allowing the user to control the movement of the 3D

objects could produce some problems regarding the UI needed for that operation and the reality sense experienced. An additional UI element would be required for the movement controls, which would further populate the screen and diminish the available space for the camera view. Furthermore, the model's movement would need to be restricted to some sections of the scene to prevent the user from guiding the 3D objects to locations impossible to access in the real world. Integrating the movement into the model's animation would also require constraints to be applied to ensure, once again, the 3D object only moves to accessible places. Although it would require extensive testing, this addition could positively impact the user experience.

- **Making the application compatible with AR glasses**

With the evolution of AR hardware, AR glasses are growing in popularity. Such devices could help in increasing the sense of reality perceived by the user when viewing virtual objects on top of the real-world scene. In the case of this dissertation's work, AR glasses would provide a more seamless experience to rock art visitors. Furthermore, it could eliminate some problems related to the lighting of the real-world environment.

- **Adding help elements to some of the pages**

The lack of UI elements containing instructions on how to perform some of the application's operations was the most noted point by test participants. UI elements such as instructions panels, hints and menus could be added to the current UI of the application, aiming to improve this aspect. Special care would be required when designing such elements to prevent overfilling the screen.

Chapter 7

Conclusions

AR presents great potential to improve the user experience in cultural scenarios. The versatility offered by this technology allows for joining the real world with virtual elements, enabling the enrichment of cultural exhibits and sites and allowing visitors to experience novel situations.

In the context of cultural visits, especially outdoor ones, AR faces several problems due to the uncontrolled nature of the environment. Apart from uncontrollable variables like the lighting of the scene and its vegetation, the distance to cultural objects can also pose an obstacle to users. These issues are encountered when using AR for rock art.

This dissertation explores the issues of applying AR to flat rock engravings and proposes methods to mitigate the problems identified. Furthermore, it investigates the feasibility of applying AR to the Côa Valley rock engravings and the ways in which this technology can be used to help visitors visualize and understand rock art, providing answers to the research questions described in section 1.2.

Regarding RQ1, the main issues with applying AR to flat rock engravings in outdoor conditions are the lack of control over the environment in which the engravings are located and the detection of such art elements from different angles and perspectives. Different approaches were developed and tested to mitigate these issues, as presented in chapter 5. Some of the tested techniques improved the robustness of the application while others mostly focused on improving its performance. Robustness improvements mostly led to decreases in performance and, therefore, a balance between these two factors was reached.

The results obtained from the user studies described in chapter 6 demonstrate the usefulness of a mobile AR application in the interpretation of rock art. With simple and intuitive operations, this application was able to aid users in visualizing and distinguishing rock engravings, answering RQ2. Furthermore, it also provided users with a wider range of information and context regarding specific engravings.

Finally, regarding RQ3, it was concluded that it is feasible to apply AR to rock art visits. Even though the AR technologies currently available lack the power to make the experience immersive and seamless on their own, this can be compensated using the methods described in chapter 5. Most users that participated in the user studies agreed that the developed application would be a

great way to complement the information provided by the tour guide and that it would improve their rock art visiting experience.

7.1 Improvements

Although the feedback received from the users was positive, there are some points that can be improved. Firstly, some users showed difficulties during the engraving detection process, mostly due to the lack of UI elements containing help instructions. This issue was also pointed out on the 3D Visualization Page, where some test participants felt a lack of helping elements. The addition of panels and hints providing instructions to the users at each step of the application could significantly improve their experience so that this aspect could be improved.

As mentioned in section 6.4.2, the 2D animations were unclear to some test participants. Displaying the animation in the AR scene poses a few problems, so it would be worth exploring ways to improve this process. Apart from that, new methods of displaying the 2D animations should also be researched, with the goal of providing users with a clear view of the animation while keeping the simplicity of the application.

It might be possible to further meliorate the engraving detection step by exploring the use of image detection algorithms based on machine learning. Although this would require a large dataset of engraving pictures, the performance of the application when detecting engravings in the camera view could be greatly boosted.

7.2 Future Work

With AR constantly evolving, there is high potential to evolve this dissertation's solution. Regarding hardware, it is worth exploring the use of LiDAR sensors to enhance the detection of engravings. These sensors are currently only available on some Apple devices, but their popularity is growing. LiDAR sensors could help detect engraving lines with high precision, enhancing the position estimates for the virtual outlines. Furthermore, the detection of surfaces in the real world could significantly benefit from these sensors' ability to determine distances between the mobile device and the objects being captured by the camera. Another interesting piece of hardware that could be integrated into this application is AR glasses. As suggested by some users during the usability testing studies, using AR glasses to render virtual objects on top of the real-world view instead of the phone's screen could positively impact the experience of rock art site visitors by removing the need to hold a device with their hands. Although this type of equipment is currently expensive and not accessible to everyone, it would present a great way to improve the developed application's performance and acceptance by the final users.

Once the solution is expanded to work with other rocks and not just Penascosa's rock number six, performance will be a key factor to consider. As demonstrated in section 5.2, the image detection performance starts to decrease as the number of reference images in the database increases. When adding support for other rocks, it will be important not to overload the reference database

with too many reference images. Instead of using the same database for all the rocks supported by the application, each rock can have its own reference image storage. In order to determine at runtime which database to use, there are two main options: 1) ask the user which rock/site is being visited and 2) use GPS data to determine the correct rock/site. The first option requires a new user interface to be developed so that users can input their location. Although it is the easiest to implement, it is prone to user mistakes and increases the number of actions users are required to take to use the application. The second option presents a more seamless way to determine the correct reference database based on the user's location context. Using the mobile device's GPS data, one is able to determine the user's position in the world. By comparing the user's position to the position of the supported rocks, the application can determine which rock is closest and load its reference database.

In a more advanced state of the application, the reference databases could also be enhanced with pictures taken by the users. This idea would require a back office to manage picture submissions, which would, in turn, increase the complexity of the application. However, allowing users to populate reference databases with pictures taken by them at the rock art sites would be a way to ensure the reference images would stay up to date, therefore increasing the robustness of the application.

Finally, the types of media supported on the Information Page could be improved. In addition to textual information and images, which are already supported, it would be worth exploring video support. Furthermore, there would be benefits to having a text-to-speech conversion module so that users can listen to the textual descriptions while viewing information in other formats, like images and videos.

References

- [1] D. W. Mizell A. L. Janin and T. P. Caudell. Calibration of head-mounted displays for augmented reality applications, 1993.
- [2] Violette Abergel, Kevin Jacquot, Livio De Luca, and Philippe Veron. Towards a slam-based augmented reality application for the 3d annotation of rock art. *The international journal of virtual reality*, 19(2), 2019.
- [3] Ronald T. Azuma. Ar - a survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [4] Ronald T Azuma et al. The challenge of making augmented reality work outdoors. *Mixed reality: Merging real and virtual worlds*, 379:390, 1999.
- [5] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, May 2009.
- [6] Mafkereseb Kassahun Bekele, Roberto Pierdicca, Emanuele Frontoni, Eva Savina Malinverni, and James Gain. A survey of augmented, virtual, and mixed reality for cultural heritage. *J. Comput. Cult. Herit.*, 11(2), mar 2018.
- [7] Silvia Blanco-Pons, Berta Carrión-Ruiz, and José Luis Lerma. Augmented reality application assessment for disseminating rock art. *Multimedia tools and applications*, 78(8):10265–10286, 2019.
- [8] Silvia Blanco-Pons, Berta Carrión-Ruiz, José Luis Lerma, and Valentín Villaverde. Design and implementation of an augmented reality application for rock art visualization in cova dels cavalls (spain). *Journal of cultural heritage*, 39:177–185, 2019.
- [9] Chien-Wen Chen, Wen-Zheng Chen, Jain-Wei Peng, Bo-Xun Cheng, Tse-Yu Pan, Hsu-Chan Kuo, and Min-Chun Hu. A real-time markerless augmented reality framework based on slam technique. In *2017 14th International Symposium on Pervasive Systems, Algorithms and Networks 2017 11th International Conference on Frontier of Computer Science and Technology 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC)*, pages 127–132, 2017.
- [10] Roger Cheong. The virtual threat to travel and tourism. *Tourism Management*, 16(6):417–422, 1995.
- [11] Paolo Clini, Emanuele Frontoni, Ramona Quattrini, and Roberto Pierdicca. Augmented reality experience: From high-resolution acquisition to real time augmented contents. *Advances in Multimedia*, 2014, 12 2014.

- [12] Riccardo De Benedictis, Carlo De Medio, Augusto Palombini, Gabriella Cortellessa, Carla Limongelli, and Amedeo Cesta. Fostering the creation of personalized content for cultural visits. *Applied Sciences*, 11(16), 2021.
- [13] Arindam Dey, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan. A systematic review of 10 years of augmented reality usability studies: 2005 to 2014. *Frontiers in Robotics and AI*, 5, 2018.
- [14] Lidia Falomo Bernarduzzi, Ester Maria Bernardi, Alberto Ferrari, and Maria Carla Garbarino. Augmented reality application for handheld devices: How to make it happen at the pavia university history museum. *Science education*, 30(3):755–773, 2021. PMID: 33758467.
- [15] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62, jul 1993.
- [16] Angel García-Crespo, Javier Chamizo, Ismael Rivera, Myriam Mencke, Ricardo Colomo-Palacios, and Juan Miguel Gómez-Berbís. Speta: Social pervasive e-tourism advisor. *Telematics and Informatics*, 26(3):306–315, 2009. Mobile and wireless communications: Technologies, applications, business models and diffusion.
- [17] Dragoş Gheorghiu, Livia Ştefan, Mihaela Moţăianu, and Geir Harald Samuelsen. *Rhythms in Stone: Revealing and Augmenting the Human Presence in Mesolithic Rock Art*, pages 349–365. Springer International Publishing, Cham, 2022.
- [18] Alexandru Gherghina, Alexandru-Corneliu Olteanu, and Nicolae Tapus. A marker-based augmented reality system for mobile devices. In *2013 11th RoEduNet International Conference*, pages 1–6, 2013.
- [19] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(6):1706–1724, 2017.
- [20] Anders Henrysson, Mark Billinghurst, and Mark Ollila. Ar tennis. <http://doi.acm.org/10.1145/1179849.1179865>, 01 2006.
- [21] Feng Huang, Yu Zhou, Yao Yu, Ziqiang Wang, and Sidan Du. Piano ar: A markerless augmented reality based piano teaching system. In *2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 47–52, 2011.
- [22] Shafaq Irshad and Dayang Awang. A ux oriented evaluation approach for mobile augmented reality applications. pages 108–112, 11 2018.
- [23] B. Jiang, U. Neumann, and Suya You. A robust hybrid tracking system for outdoor augmented reality. In *IEEE Virtual Reality 2004*, pages 3–275, 2004.
- [24] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. page 85 – 94, 1999. Cited by: 1741; All Open Access, Green Open Access.
- [25] Anabel Kečkeš and Igor Tomicic. Augmented reality in tourism - research and applications overview. *Interdisciplinary Description of Complex Systems*, 15:158–168, 06 2017.

- [26] Soo Kyun Kim, Shin-Jin Kang, Yoo-Joo Choi, Min-Hyung Choi, and Min Hong and. Augmented-reality survey: from concept to application. *KSII Transactions on Internet and Information Systems*, 11(2):982–1004, February 2017.
- [27] P. Lang, A. Kusej, A. Pinz, and G. Brasseur. Inertial tracking for mobile augmented reality. In *IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276)*, volume 2, pages 1583–1587 vol.2, 2002.
- [28] Nanxi Li, Chong Pei Ho, Jin Xue, Leh Woon Lim, Guanyu Chen, Yuan Hsing Fu, and Lennon Yao Ting Lee. A progress review on solid-state lidar and nanophotonics-based lidar sensors. *Laser & Photonics Reviews*, 16(11):2100511, 2022.
- [29] Nanxi Li, Chong Pei Ho, Jin Xue, Leh Woon Lim, Guanyu Chen, Yuan Hsing Fu, and Lennon Yao Ting Lee. A progress review on solid-state lidar and nanophotonics-based lidar sensors. *Laser & Photonics Reviews*, 16(11):2100511, 2022.
- [30] Lena Jingen Liang and Statia Elliot. A systematic review of augmented reality tourism research: What is now and what is next? *Tourism and Hospitality Research*, 21(1):15–30, 2021.
- [31] Jacob B. Madsen and Claus B. Madsen. Handheld visual representation of a castle chapel ruin. *J. Comput. Cult. Herit.*, 9(1), nov 2015.
- [32] Goodman Gwasira Martha Mosha, Tilman Lenssen-Erz. The brandberg augmented reality experience: Past and present realities. In *Second African Conference for Human Computer Interaction: Thriving Communities*.
- [33] Giorgos Tsigkas Nikou, Giorgos Sfikas, Anastasios Pasialis, Andreas Vlachopoulos, and Christophoros. Markerless detection of ancient rock carvings in the wild: rock art in vathy, astypalaia. *Pattern Recognition Letters*, 135:337–345, 2020.
- [34] Gerhard Reitmayr and Tom W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 109–118, 2006.
- [35] M. Ribo, P. Lang, H. Ganster, M. Brandner, C. Stock, and A. Pinz. Hybrid tracking for outdoor augmented reality applications. *IEEE Computer Graphics and Applications*, 22(6):54–63, 2002.
- [36] Sofia Alexandra Gonçalves Rodrigues. *The Augmented Reality Systems in a Maintenance of Equipment tutorial context*. Thesis, 2015.
- [37] Aliisa Råmark. *Rock Art in Technicolour Exploring Augmented Reality as an Accessibility Tool for Rock Art in Tanum World Heritage*. Thesis, 2022.
- [38] Pawan K. Shrestha, Matt J. Pryn, Jia Jia, Jhen-Si Chen, Hector Navarro Fructuoso, Atanas Boev, Qing Zhang, and Daping Chu. Accommodation-free head mounted display with comfortable 3d perception and an enlarged eye-box. *Research*, 2019, 2019.
- [39] D. Stricker and T. Kettenbach. Real-time and markerless vision-based tracking for outdoor augmented reality applications. In *Proceedings IEEE and ACM International Symposium on Augmented Reality*, pages 189–190, 2001.

- [40] B. Thomas, B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, and W. Piekarski. Arquake: an outdoor/indoor augmented reality first person application. In *Digest of Papers. Fourth International Symposium on Wearable Computers*, pages 139–146, 2000.
- [41] Hideaki Uchiyama and Eric Marchand. Object Detection and Pose Tracking for Augmented Reality: Recent Approaches. In *18th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, Kawasaki, Japan, February 2012.
- [42] A. Vlachos, M. Perifanou, and A. A. Economides. Augmented reality applications for urban cultural heritage sites: An overview. In *2022 International Conference on Advanced Learning Technologies (ICALT), 1-4 July 2022*, 2022 International Conference on Advanced Learning Technologies (ICALT), pages 322–4. IEEE. 21970570 handheld devices camera-based devices tourism tracking methods augmented reality projects software tools urban cultural heritage sites.
- [43] Vassilios Vlahakis, John Karigiannis, Manolis Tsotros, Michael Gounaris, Luis Almeida, Didier Stricker, Tim Gleue, Ioannis T Christou, Renzo Carlucci, Nikos Ioannidis, et al. Archeoguide: first results of an augmented reality, mobile computing system in cultural heritage sites. *Virtual Reality, Archeology, and Cultural Heritage*, 9(10.1145):584993–585015, 2001.
- [44] Meredydd Williams, Kelvin K. K. Yao, and Jason R. C. Nurse. Toarist: An augmented reality tourism app created through user-centred design. *CoRR*, abs/1807.05759, 2018.
- [45] Zornitza Yovcheva, Dimitrios Buhalis, and Christos Gatzidis. Engineering augmented tourism experiences. In Lorenzo Cantoni and Zheng (Phil) Xiang, editors, *Information and Communication Technologies in Tourism 2013*, pages 24–35, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [46] Iman Zendjebil, Fakhr-Eddine Ababsa, Jean-Yves Didier, Jacques Vairon, Luc Frauciel, Martin Hachet, Pascal Guitton, and Romuald Delmont. Outdoor augmented reality: State of the art and issues. 04 2008.
- [47] Qi Zhang, Xiaoyang Zhu, Haitao Yu, and Yongshi Jiang. Enhancing rock painting tour experience with outdoor augmented reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 118–121, 2019.