AUGMENTED REALITY APPLICATION ASSESSMENT FOR DISSEMINATING ROCK ART

Present/permanent address:

Photogrammetry & Laser Scanning Research Group (GIFLE), Department of Cartographic Engineering, Geodesy and Photogrammetry, Universitat Politècnica de València, 46022 Valencia, Spain.

Author names and affiliations:

Silvia Blanco-Pons
Universitat Politècnica de València, Photogrammetry & Laser Scanning Research Group (GIFLE), Department of Cartographic Engineering, Geodesy and Photogrammetry
silblapo@doctor.upv.es
ORCID: 0000-0003-0764-2199 Telephone number: +0034 679 32 93 88

Berta Carrión-Ruiz
Universitat Politècnica de València, Photogrammetry & Laser Scanning Research Group (GIFLE), Department of Cartographic Engineering, Geodesy and Photogrammetry
bercarru@doctor.upv.es
ORCID: 0000-0001-5965-8833

José Luis Lerma
Universitat Politècnica de València, Photogrammetry & Laser Scanning Research Group (GIFLE), Department of Cartographic Engineering, Geodesy and Photogrammetry
jllerma@cgf.upv.es ORCID: 0000-0001-9443-9214

Corresponding author:

Silvia Blanco-Pons, silblapo@doctor.upv.es

Abstract Currently, marker-based tracking is the most used method to develop augmented reality (AR) applications (apps). However, this method cannot be applied in some complex and outdoor settings such as prehistoric rock art sites owing to the fact that the usage of markers is restricted on site. Thus, natural feature tracking methods have to be used. There is a wide range of libraries to develop AR apps based on natural feature tracking. In this paper, a comparative study of Vuforia and ARToolKit libraries is carried out, analysing factors such as distance, occlusion and lighting conditions that affect user experience in both indoor and outdoor environments, and eventually the app developer. Our analysis confirms that Vuforia's user experience indoor is better, faster and flicker-free whether the images are properly enhanced, but it does not work properly on site. Therefore, the development of AR apps for complex outdoor environments such as rock art sites should be performed with ARToolKit.

Keywords Archaeology, Augmented reality (AR), Mobile application (app)**,** Markerless tracking, ARToolKit, Vuforia

1 Introduction

Augmented Reality (AR) refers to the computer device visualisation technology that allows the user to combine information coming from both the real world and the virtual world in real time. Nowadays, one of the most common ways of using AR is through either smartphones or tablets. These portable devices are highly extended worldwide and integrate one or two digital cameras. Smartphones and tablets can be considered the most suitable devices nowadays to implement visualisation technologies. Moreover, it is possible to add in its real position [1] any virtual content such as 3D models, 2D images or text information on top of the scene captured by the smartphone/tablet camera since the device position is calculated in real time to place the virtual content properly. Simultaneous visualisation of both real and virtual objects helps the users to understand quickly the added information and contributes to enhance their experiences.

Smartphones are fully equipped with a wide range of sensors such as Global Navigation Satellite Systems (GNSS) (namely GPS and Glonass), Bluetooth, high-resolution video camera, accelerometers and gyroscopes, and wireless technology ready to connect to internet e.g. through 4G/5G, among other common components. Due to this hardware evolution, the number of platforms for developing mobile AR apps is steadily increasing [2].

AR technology is the result of many years of evolution. As Carmigniani et al. [3] pointed out, AR technology already has some years of evolution and improvement. The beginnings of this technology date back to the 1950s, when Morton Heilig, a cinematographer, was interested in enriching the cinema experience and from his vision, he built “sensorama”. Throughout all these years, this technology has been changing and improving and many authors have given their own AR definition.

According to [4], AR places virtual elements in the camera scene and allows the user to see the real world, with virtual objects superimposed supplementing reality. AR is a very useful information tool, in which digital images, computer graphics and multimedia elements can be combined in an interactive way [5]. Moreover, Azuma et al. [6] established the essential features of an AR system: it combines real and virtual objects in a real environment, in real time and registers real and virtual objects with each other.

Nowadays, there are many AR apps in all sorts of fields, such as medicine, manufacturing, education, robot path planning, military aircraft, entertainment and even tourism. Moreover AR provides a new way to show the cultural heritage that provides the user enriching experiences [7, 8]. In recent years, many AR apps are developed especially on mobile apps to disseminate cultural heritage. Gutierrez et al. [9] described an example of app to show and disseminate cultural heritage about historical graffiti in the Temple of Debod. Ridel et al. [10] applied a projected 3D visualisation to a physical archaeological artefact that highlights its features. AR was also used by [11] for enhancement and visualisation of a wooden sculpture. Visualisation of 3D models in-situ is another type of AR app that offers the chance to the user to observe archaeological findings [12].

These types of apps in the archaeological museums field have become very popular in recent years due to the fact that the visit is highly enriched [13]. However, when AR apps are used to visit rock art sites, it is necessary to take into account special (singular) features of this type of environment. Generally, rock art sites are located in remote areas constantly exposed to direct sun light and different altering agents are gradually threatening and degrading the art [14, 15]. This means that there are sites with very poor coverage, limited (if any) broadband signal and heterogeneous lighting where high contrasts can be found especially in shelters. In addition, due to the fragility of conservation and the similarity between the paintings and the rock that supports them, it is extremely difficult in most of the cases to recognise painted features. On the other hand, it should be noted that the visitor must be a minimum distance away from the motifs; there is usually a minimum and maximum distance rate to visualise the paintings. The main motivation behind this work is to find an AR library that works fine in these rough and harsh environments, allowing the development of future apps to disseminate the rock art scenes full of faint motifs.

Several libraries can be found to develop AR apps running with markers so that the camera on a mobile device can recognise an image pattern and project the virtual object over the marker. The literature reports numerous research on the factors that can affect the markers recognition. Khan et al. [16] analysed eleven factors that affect the marker quality and evaluated the effects of each factor on the quality of the marker tracking system. Also, several studies [17, 18] analysed other factors such as distance between marker and device camera, number of markers, shape and size of the markers or lighting conditions; other authors published a comparison of several libraries for AR implementation [5, 19]. However, there is a lack of comparative user and developer assessment reporting the experience of using AR apps to disseminate complex cultural heritage sites.

The final goal of our research project was to create an AR app able to show the user a recreation and interpretation of the paintings found in a prehistoric rock art site. In order to achieve this**,** this research assesses the factors affecting the user's experience in two scenarios: 1) indoor, trying to test its usefulness for purpose in either a museum, exhibition, gallery or laboratory, making use of a printed colour paper of a rock art scene; and 2) outdoor on site in a rock art cave (shelter). In addition, the assessment can be used by the developers to confirm which library is expected to run better in both scenarios. Hereby, the aim of this paper is to assess the implemented rock art app on prehistoric rock art features, using the Unity3D engine platform, and both ARToolKit [20] and Vuforia [21] libraries.

The paper is structured as follows. Section 2 tackles the subject of AR for rock art sites and both ARToolKit and Vuforia graphic libraries are presented with emphasis on visual-based tracking. Section 3 explains the design and implementation of the AR rock art app, considering two environments, one indoor and another outdoor. Section 4 presents the results achieved with both libraries. Section 5 discusses the results and summarises them. Finally, Section 6 draws some conclusions.

2 AR and tracking methods

2.1 Context

In this study, we are focused on AR apps to improve both the knowledge and the dissemination of prehistoric rock art sites. The motifs represented in rock art are usually rather difficult to visualise without any help from archaeological experts because the paintings are severely damaged. Moreover, the paintings are usually located in open-air sites. AR rock art apps can show the current state of the paintings or even a simulation of the real paintings, making easier the understanding of the represented motifs. Furthermore, these apps should run on smartphones and tablets, so they are conceived to be used *in situ* under a wide range of unexpected circumstances.

In order to achieve a spatially correct overlap of virtual information, the main AR challenge is the tracking of the users' camera: the virtual content must be perfectly aligned with the real world. Therefore, it is necessary an accurate estimation of the position and orientation of the digital camera [22]. The objective of the tracking system is to determine the pose of the camera in real time. Whenever the user moves the AR device, the tracking system recalculates the new pose in real time and thus the virtual contents have to remain aligned with the real object. The camera pose is calculated with six degrees of freedom, three translation parameters x, y, z and three orientation parameters yaw, pitch, roll [23].

The proper visualisation of virtual contents depends on the tracking method. Vision-based tracking is a widely used AR method for camera tracking. This method calculates the camera pose from the data received by the camera view. The six degrees of freedom are calculated from the known geometric features of the objects and their 3D geometric relations. Depending on the feature type, vision-based tracking is categorised into marker-based tracking -i.e. detection of a known pattern- and markerless -i.e. detection of natural features in a scene [24].

2.2 Tracking methods

2.2.1 Marker-based tracking

Fiducial Marker-Based Tracking [5], mostly known as marker-based tracking, is currently the most used tracking method. This method makes use of markers easily recognisable. Typical markers are black and white squares with high contrast, which are placed in the scene to be augmented. The main advantage of this marker-based tracking is that quickly recognises the markers because of the high contrast between the markers and the environment.

Both ARToolKit and Vuforia have predefined square markers (Fig. 1), but they allow the user to design their own markers. In ARToolKit, markers must be squares with continuous edges. In Vuforia, the latest release includes VuMarks which allows the users to create completely customised branding design marks [25].

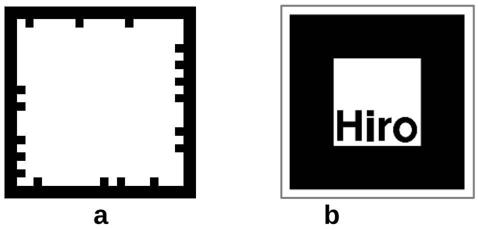

Fig. 1 Visual markers already implemented in Vuforia (a) and ARToolKit (b) libraries.

Partial marker occlusions originate the main problem in visual-based tracking methods. If the marker is partially obscured, the library will not recognise the marker, and therefore, the AR app will become unstable. Other factors that can affect the correct marker detection are, for instance, the marker size, the camera-marker distance or the scene lighting.

In some cases, marker-based tracking cannot be used owing to the fact that placing (fixing) markers in protected places or outdoor environments is restricted or not feasible. Therefore, other tracking method is required to deal with such afore-mentioned cases.

2.2.2 Markerless tracking

Scene recognition without markers is also employed with AR technologies; it is applied to facial recognition, gesture recognition and image recognition. Markerless tracking requires high computational cost, as well as demanding and complex processing to recognise enough matching features to enable pose estimation and tracking.

Markerless tracking can be classified in two methods: feature-based and model-based tracking. Feature-based tracking consist of finding natural features in the scene. Model-based methods make use of 3D model to estimate the camera pose; this method is commonly used for tracking 3D objects without texture [26]. In this paper, we focus on tracking 2D features due to several reasons: 1) rock art paintings are usually located in protected archaeological areas, which must not be altered in any way and hence markers are not allowed to be placed on site; 2) the scene is full of texture and details; and 3) the motifs in the shelter are split in small units (panels) that can usually be assumed as flat.

2.2.3 Feature-based tracking

Also called Natural Feature Tracking (NFT) [23, 27], this method only uses natural features easily detectable in the scene such as edges or corners; hence, this is a nonintrusive approach. The scene to be recognised must be rich in details and have preferably high contrast. It is necessary to obtain a sufficient number of features in order to determine correctly the camera pose.

Figure 2 presents an overview of the NFT method. This method requires a known reference image which is computed offline before the tracking to extract the keypoints and describe the feature vectors; these data is stored in the feature database. Then, the keypoints of each acquired image in real time (with the mobile device) are extracted and described, and these feature vectors are matched with those of the feature database. Finally, the camera pose is obtained with this set of correspondences, after applying homography between pairs of views (reference image and acquired images) [28].

There are several types of feature detectors and descriptors aimed at identifying keypoints and the calculation of feature vector that fully describes the keypoint and its local neighbourhood [29, 30]. Once keypoints are located (with the feature extraction), a vector that describes the feature of a keypoint is computed (with the feature description).The most popular descriptor is SIFT [31] but there are other such as SURF, FAST, ORB, PCA-SIFT, ASIFT described in [28–30, 32]. ARToolKit uses the FREAK descriptor, which is faster to compute with lower memory load and more robust than other descriptors [33].

Large errors in pose determination cause incorrect alignment of the virtual content. This is one of the challenges, because rock art paintings do not usually have high contrast and there are no large colour differences in the scene (Fig. 3a). Nevertheless, user is excited once he/she is able to visualise the motifs in situ, because they are hard to visualise in a particular moment (during the visit to the archaeological site). Therefore, the relevance here is to get the experience of getting the motifs on; the relevance of aligning the motifs perfectly is drastically reduced. Once the visitor knows what to look for in a particular panel, it is easier to identify the paintings under real conditions.

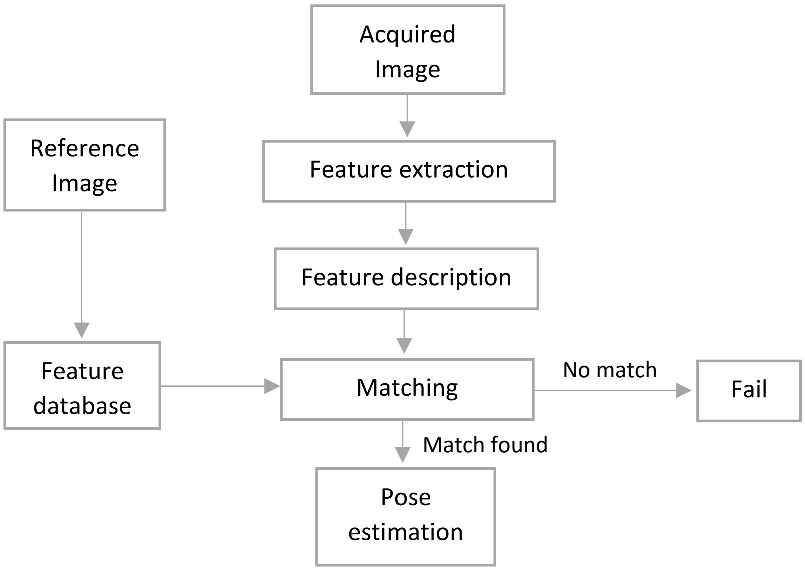

Fig. 2 Overview about the natural feature tracking system.

2.3 Libraries

In order to achieve our goal, the AR app will be used *in situ* hence it must work on mobile devices. One of the quickest and easiest ways to develop a cross-platform app is using Unity software game engine [34], although other alternative exists such as Unreal Engine

[35] and CryENGINE [36]. Unity is one the most used game engines, allows the user to work easily with 2D images and 3D models, and is compatible with all mobile platforms.

AR libraries provide facilities for markers recognition and tracking, such as 2D features and even 3D objects. It is possible only with the library components to make a basic AR app. In recent years, the number of AR libraries has increased greatly thus, there is a wide range of them. For our approach, the library should meet the following requirements:

- Open source or free license.
- NFT or image recognition.
- Plugin for Unity.
- Complete documentation.

Currently there are a large number of AR libraries; ARTag [37], ArUco [38] and DroidAR [39] are examples of AR marker-based libraries; Metaio, Vuforia, Wikitude, D'Fusion, ARToolkit and ARmedia are studied in [5], but many libraries disappeared or became obsolete over time, such as Metaio that was acquired in 2015 by Apple. After analysing the current AR libraries, it can be stated that the most used libraries meeting our requirements nowadays are both Vuforia and ARToolKit.

2.3.1 Vuforia

Vuforia is probably the most used AR platform. Currently it is not free, but it has a free license for apps in development, thus all its facilities can be used and tested. Vuforia provides great recognition capabilities based on image recognition, such as Image Target by Vuforia. Vuforia detects natural features in an image and compares them with a database of known target resources. In order to create this database, Vuforia provides a web service (*Vuforia Target Manager)* to upload the Image Target for processing and evaluation. Image Target is evaluated with stars based on the features detected. Whenever the Image Target does not display stars is because there are not detected enough features for image tracking. Therefore, recognition and tracking are both impossible within this image. On the contrary, an Image Target evaluated with five stars means that markerless tracking is perfect for this image [40]. Furthermore, Vuforia can be used with Android, iOS, UWP mobile systems and digital eyewear; also it can also be built with Unity software.

2.3.2 ARToolKit

Hirokazu Kato developed ARToolKit, the most popular library for AR app, in 1999. Two years later, the Human Interface Technology Laboratory (HIT Lab) of Washington University supported it. Next in 2015, it was acquired by DAQRI, who released all ARToolKit features, such as natural feature tracking libraries for iOS, Android and Unity versions. It is available free and open source under a LGPL v.3 License. Since then, ARToolKit is maintained by this company and they will soon release version 6 with new feature [41]. Therefore, ARToolKit is currently a well-documented open source library where the code can be studied and modified.

ARToolKit includes NFT for image recognition, just like Vuforia; the first step is to generate an ARToolKit NFT dataset from the digital image, with the features detected. For this purpose, ARToolKit provides *genTexData* tool, which generates the necessary files for tracking. First, the source image is resampled at multiple resolutions, generating an image set (.iset) file. Then, the detected features are saved in two files (.fset and .fset3): .fset contains the features used in continuous tracking, and .fset3 contains the features used for identifying the pages and initialising the tracking [42].

ARToolkit provides a camera calibration service, a cloud-and-crowd-based solution for generating and downloading calibration parameters for different smartphones [43]. Nevertheless, in order to achieve accurate results, it is important to calibrate each camera correctly for each camera resolution. To carry out the camera calibration, ARToolkit provides an Android app on Google Play called *ARToolKit Camera Calibrator*. In this study, all three smartphones have been calibrated for image resolutions of 320x240 and 720x480 pixels.

3 Design and implementation

The great challenge in AR apps is to recognise enough features in a real environment to determine the pose and orientation of the camera in real time. Therefore, the tracking image should have a great number of edges and surface details. This is quite difficult in rock art motifs because rock art scenes are faint, poor in details and there is no colour contrast. Thus, to study this problem in depth, some rock art scenes have been analysed in the Cova dels Cavalls (Fig. 3), where the whole scene is in red and brownish colours.

In this study, ARToolKit and Vuforia libraries have been evaluated in order to know which is best to develop AR apps dealing with Levantine rock art motifs. Several experiments have been performed to analyse the user experience in both indoor and outdoor environments. Users are usually annoyed of apps that delay running too much (a few second holding in front of the motifs). That is why, key factors have been selected for estimating the user's experience (vid. Section 3.2). For each factor, a different experimental setup has been designed. As stated before, these factors are evaluated in two scenarios, one indoor and another outdoor in order to cover a wide range of applications from the same smartphone app.

3.1 Case study: Cova dels Cavalls

The Cova dels Cavalls is located in Castellón (Spain), in the Valltorta ravine. It was one of the most important discoveries at the time because of the quantity and variety of its rupestrian manifestations. However, nowadays the paintings are faint, highly deteriorated and it is difficult to recognise them [44].

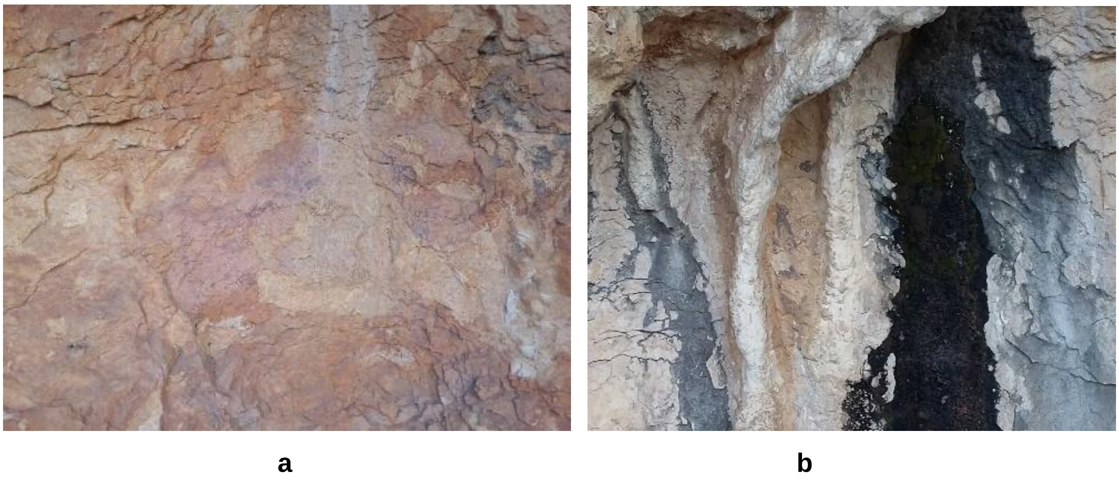

Fig. 3 Cova dels Cavalls: **a** Scene 1: Hunting scene; **b** Scene 2: Overview of motifs found on Shelter 2.

3.2 App requirements

There are some indispensable requirements that our AR app must meet in order to achieve a proper user experience. The most important ones are summarised next:

- Keypoints identification in the rock art scenes.
- Positioning of the 2D image (virtual content) at the right target in the real world.
- Quick computation of the pose estimation (tracking process).
- Working in different lighting conditions

3.3 Evaluated factors

Taking into account the apps requirements, we have evaluated a list of factors using both libraries in different smartphones: 1) Samsung Galaxy 4; 2) Samsung A3; and 3) BQ Aquaris E4. The factors evaluated are described next.

3.3.1 Feature database

The image tracking will be possible if the reference image contains enough distinctive features. For these reason, the first step is the generation of the feature database with the keypoints extracted from the reference image. We are going to assess the number of keypoints generated to run the NFT.

3.3.2 App initialisation time

The time elapsed from pressing the icon on the screen until the app is ready to run depends on the app initialisation time and data loading. The AR initialisation takes a while; an AR app usually has heavy data files such as 3D models, 2D images and databases for tracking. Hence, the initialisation time can last long depending on the number and type of data so this waiting time might confuse the user unless he/she is not alerted.

3.3.3 Tracking time

The tracking time is the processing time required to calculate camera pose and show the virtual information on the display. The processing amount of time depends on the type and number of descriptors used and can vary with the distance. Although we know that ARToolKit uses FREAK descriptor, it is not possible to know which descriptor Vuforia integrates because this library is not open source and this information is a kind of black box. Anyway, the tracking time will be measured using both libraries in order to know which is faster. This factor is evaluated indoor and outdoor.

3.3.4 Occlusion percentage

If the scene is not fully visible, the recognition may not work properly because the tracking system requires enough features to obtain the pose estimation. In a partially occluded scene, the tracking may be lost or the virtual image jittering may increase. The occlusion percentage is also tested with an image.

3.3.5 Lighting conditions

In real outdoor environments, lighting conditions cannot be controlled hence, according to the day time and the weather conditions, the sun's angle can affect the visualisation and tracking. Images with different exposure time are printed to evaluate this factor.

4 Experimental results

4.1 Indoor AR app dealing with printed colour image of a rock art scene

First, both libraries were assessed in laboratory recreating similar scene content that a user would have found in a real environment. Thus, a photograph of a hunting scene of "La Cova dels Cavalls" had been printed in colour. It is worth noting that the original image had not enough contrast to highlight the paintings from the support rock (Fig. 4a). Therefore, this input image was enhanced to emphasize paintings (Fig. 4b). The printing area was 28 x 19 cm (Din A4 page size) and the tests were undertaken after focusing the smartphone’s camera towards the photograph presented in Fig. 4b.

Generally, the rock art paintings are heavily deteriorated hence its interpretation is difficult. In addition, the lighting effects can complicate the reading e.g. if there is direct sunlight. Thus, it is usually difficult to guess or infer what the drawn figure represents. The indoor AR app shows a simulation of the original state of the paintings in the smartphone camera view (Fig. 5); the rightness of the simulation is not discussed here. It is display to clarify citizens how it might seem in the past during the Neolithic period. With the developed AR app, visitors can quickly visualise and understand the rock art paintings, i.e. the scene with the motifs.

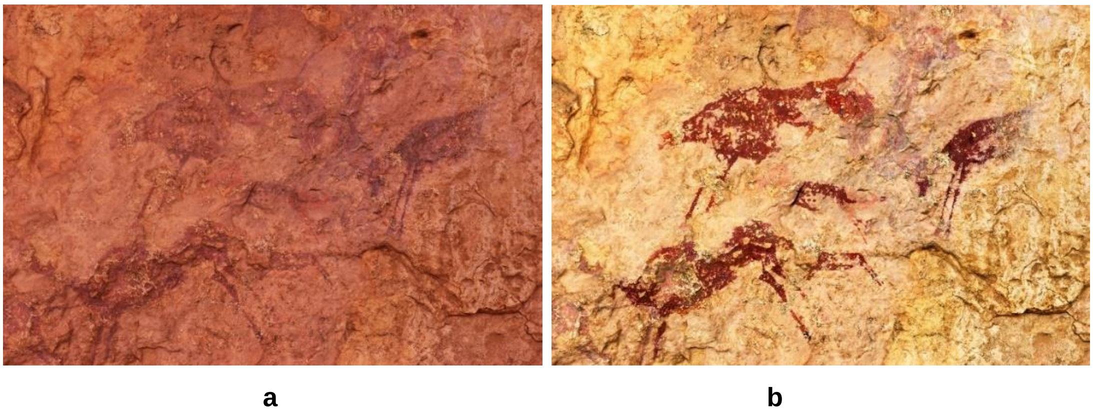

Fig. 4 a Input image taken with a smartphone in Cova dels Cavalls. **b** Enhanced image used to track the corresponding hunting scene.

4.1.1 Features database

Both tested libraries provide an offline method to generate the database with the extracted keypoints.

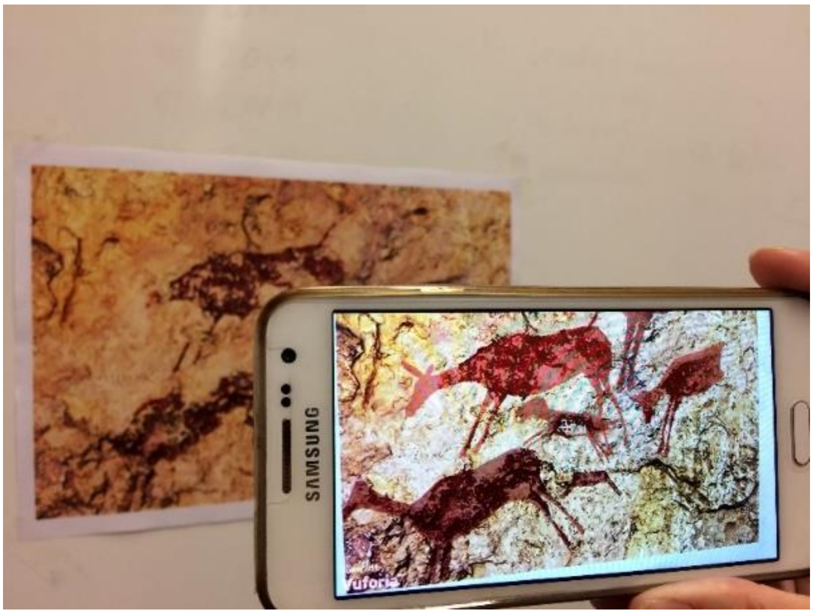

Fig. 5 Indoor testing setup, using a smartphone in front of the printed image.

ARToolKit provides the *genTexData* program to generate the features data files. This program allows users to select the level of feature extraction for tracking and initialisation, being 0 the minimum level and 4 the maximum. This step determines the amount of extracted features. In this case, the maximum level was specified to achieve more tracking features owing to the fact that the study scene is far from ideal. Therefore, a great number of known features were obtained in the database; 9111 keypoints fset and 765 keypoints fset3.

On the other hand, with *Vuforia Target Manager* was generated the database for the tracking, and the image used was evaluated with five stars, which means that the features database contains a large number of keypoints (Fig. 6).

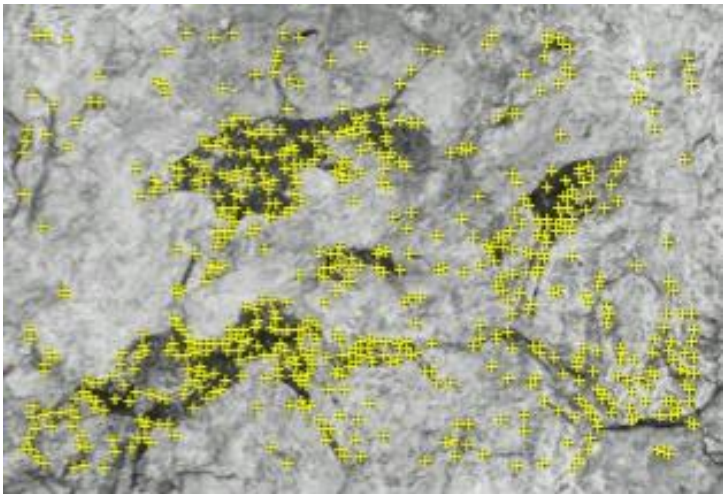

Fig. 6 Features detected in the database by Vuforia.

4.1.2 App initialisation time

Trying to determine how long the user had to wait to use the app, the initialisation time was measured (Fig. 7). The initialisation time takes longer with ARToolKit than with

Vuforia. Regarding the smartphone used, the fastest smartphone was the Samsung A3, and the slowest the BQ Aquaris E4.

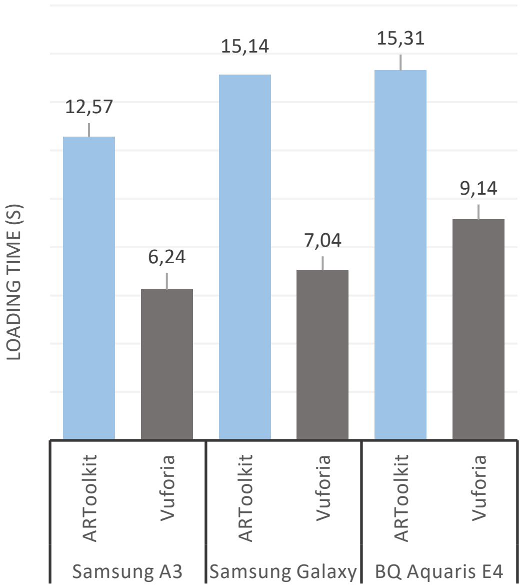

Fig. 7 Time taken to run the AR app on the three smartphones.

4.1.3 Tracking time

Tracking time was measured in each of the three smartphones used in this study at different distances of the image. The minimum distance employed to begin the test was 20 cm and then the tracking time was measured every 10 cm up to 100 cm.

In all Samsung A3 cases, the time measured was less than 5 s with the ARToolKit library, with the exception of distances equal or longer than 90 cm (Fig. 8). Therefore, that distance was considered the maximum distance that the app supports for this smartphone. From 90 cm, the app was not running, that is, stops loading the virtual information. It is also interesting to highlight that the lowest loading time was obtained at 80 cm.

Concerning the Samsung Galaxy, generally lower ARToolKit loading times are obtained than with the Samsung A3 times (Fig. 8). In most cases, the times are less than 2 s and the lowest time (1.20 s) has also being obtained at 60 cm. On shorter distances, loading times are higher even though they remain below 4 s. On the other hand, at distances longer than 95 cm, the app stops loading the virtual information.

For all experiments performed with the BQ Aquaris E4 (Fig. 8.), loading time with ARToolKit is less than 3 s; the lowest time is 1.20 s and again at a distance of 80 cm. Furthermore, 95 cm is considered the maximum distance that the app supports in this smartphone.

On the other hand, the loading time achieved with Vuforia library is kept constant (0.35 s) and minimum independently of the smartphone used (Figs. 8, solid line). It can be reported that with Vuforia the tracking computation is faster, almost instantaneous.

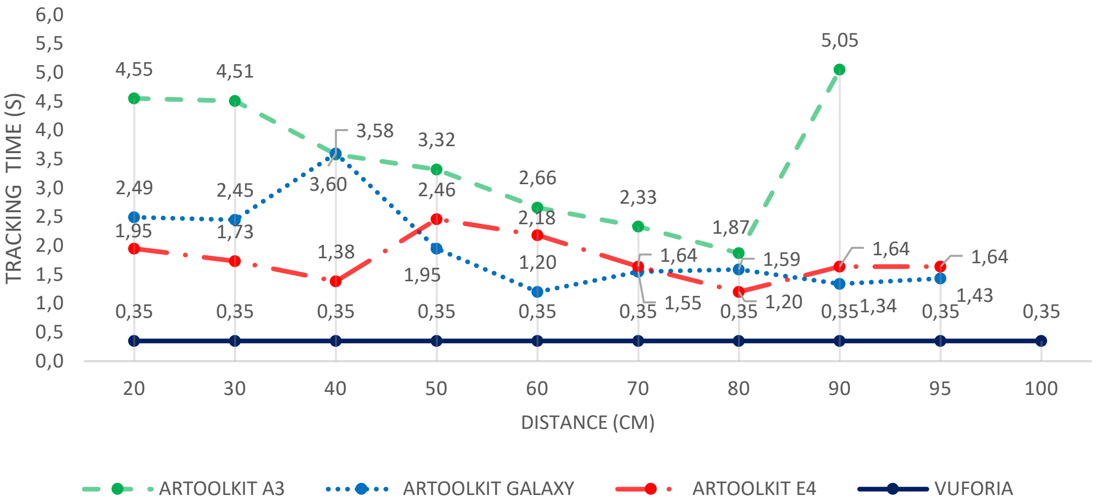

Fig. 8 Samsung A3, Samsung Galaxy and Aquaris E4 virtual information loading times at different distances; no differences were found among smartphones with Vuforia.

4.1.4 Occlusion percentage

Some parts of the image are deliberately hidden, in a way that a percentage of the image is visible and another part is completely occluded. Three occlusion percentages were applied: 25%, 50% and 75%.

With both libraries, the tracking time does not vary when a part of the image is hidden (Fig. 9). Thus, in this experimental case, the occlusion percentage does not affect the virtual content loading time.

Moreover, all the loading times are again minimum and flat with Vuforia library (Fig. 9) on Samsung Galaxy A3, as well as less than 1 s (in particular 0.32 s).

4.1.5 Lighting conditions

In order to test changes in the ambient illumination, a simulation was also prepared in laboratory. Thus, the changes that the paints suffer in their real environment due to changes in lighting have been simulated varying the exposure time. For this purpose, the exposure time of the input image (in RAW mode) is intentionally modified with Adobe Photoshop software. Once saved the image with each varying exposure, the images are printed in colour.

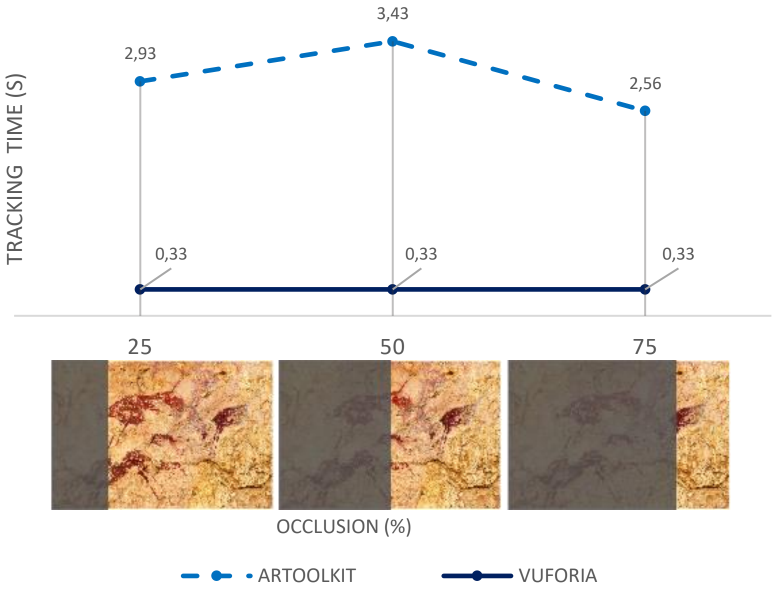

Fig. 9 Samsung Galaxy A3 loading times with different occlusion percentages: left 25%, centre 50% and right 75%.

Figure 10 shows the exposure values used and the tracking times obtained for each image exposure. Using ARToolKit, the tracking times increases a lot as the exposure time is balanced away from the original (ideal) exposure. However, using Vuforia, tracking times were kept minimum and flat (0.33 s) in all cases except the first one (extremely under exposure image).

4.2 Real environment results

Both libraries are tested on site under real environment conditions (Fig. 11) where there are problems such as surface roughness, lighting variations due to sun or even a wide range of camera-object distances. In addition, the real environment cannot be considered as a flat surface whether the whole cave is considered, although the rock art paintings used in this study belong to a panel, as a small area (within a shelter) they can be assumed as laying on a 2D plane. Nevertheless, this might cause failures during the tracking and this feature is exactly what it is going to be tested next.

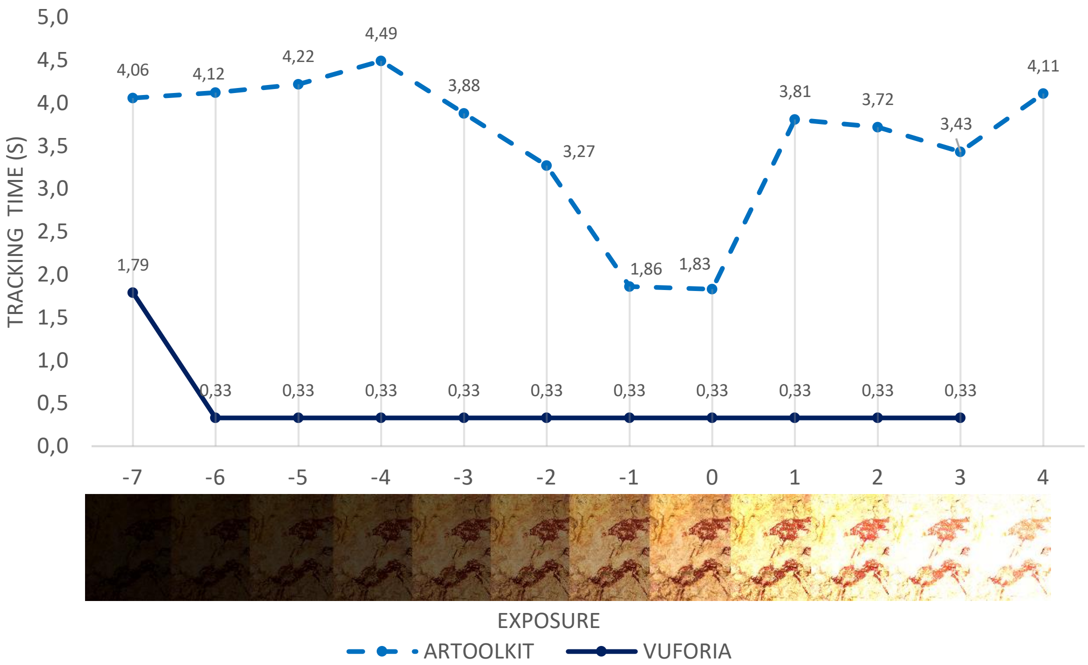

Fig. 10 Galaxy A3 loading times with different exposure times.

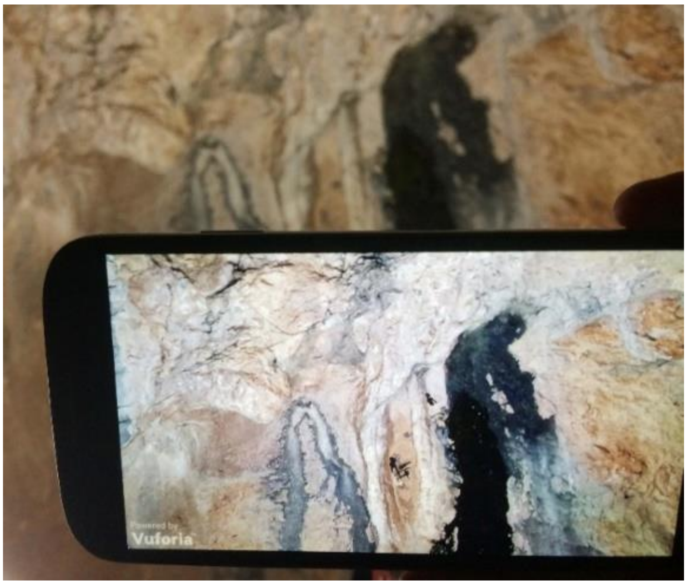

Fig. 11 Typical scenario in situ at the Cova dels Cavalls: camera in front of one of the shelters.

4.2.1 Features database

Different images were taken in the Cova dels Cavalls to test both libraries in an outdoor environment. These images were analysed with the tool *Vuforia Target Manager* (Figure 12) and ARtoolkit *genTexData* to extract the features and generate the database. While ARToolKit tool extracted a lot of features in each image, Vuforia only extracted enough features for one single image (Figure 12b), the other images were not valid for developing an AR rock art app with Vuforia library (Figure 12a and 12c).

4.2.2 Tracking time

Tracking time is measured in situ, consequently, five positions have been established at different distances from the scene. The five established positions are shown in Figure 13 and are summarised next:

- Position 1: At a distance of 50 cm from the scene. The visitors cannot be placed; it is restricted to technicians and museum staff members (Fig. 13).
- Position 2: At a distance of 100 cm from the scene. Visitors are not authorised to be in this position but they can reach this position if they lean forward and extend their arms from the walkway (Fig. 13).
- Position 3: At a distance of 150 cm from the scene. This position corresponds to the end of the walkway (Fig. 13).
- Position 4: At a distance of 200 cm from the scene. Position in the centre of the walkway (Fig. 13).
- Position 5: At a distance of 250 cm from the scene. Position at the outermost part of the walkway, next to the railing (Fig. 13).

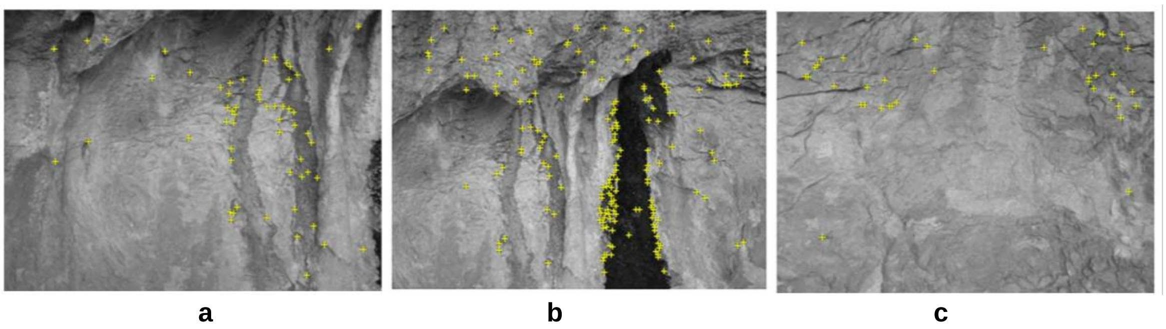

Fig. 12 Example of features detected with Vuforia in different scenes of the Cova dels Cavalls: a) Image target scored with 0 stars; b) Image target scored with 4 stars; c) Image target scored with 0 stars.

The minimum detectable distance for tracking in ARToolKit was 50 cm (Position 1); in Vuforia, 100 cm (Position 2). However, Positions 1 and 2 were discarded because these positions are not accessible for visitors. Regarding tracking times measured at Positions 3, 4 and 5, a difference is observed between the two libraries used (Fig. 14). Vuforia was always faster than ARToolKit and stable, 0.2 s. The tracking time was nearly immediate in any of the smartphones employed.

The tracking times measured with ARToolKit library were around 2 s in almost all positions, although there were some differences depending on the smartphone used. In this case, the tracking times obtained range between 1.5 and 2.5 s (Fig. 14).

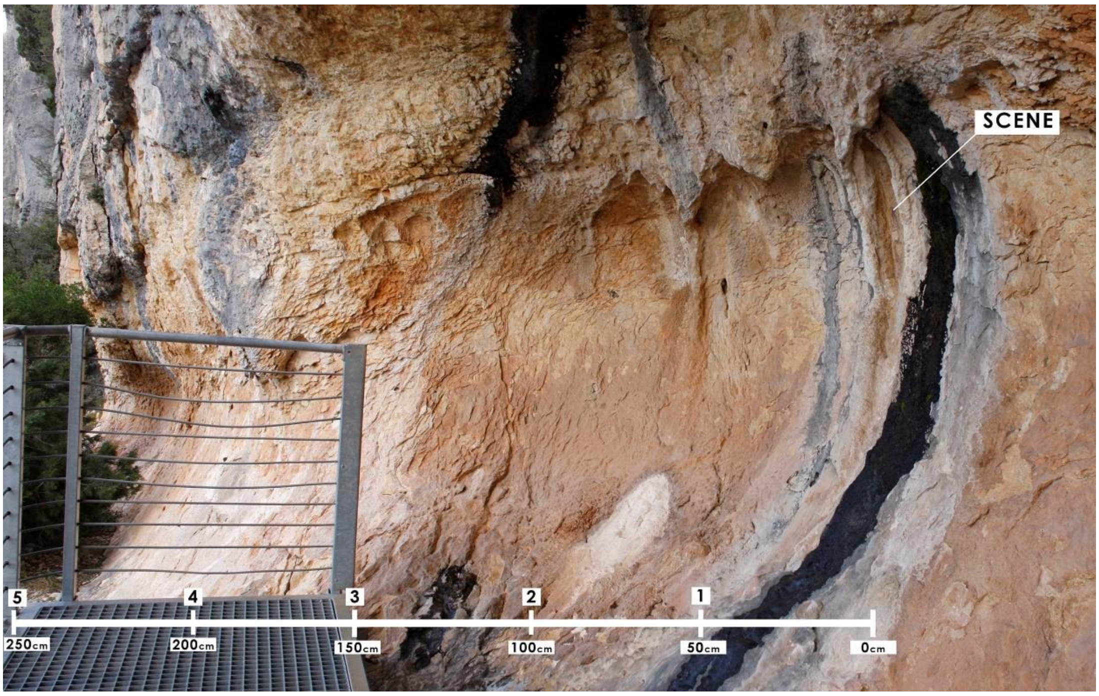

Fig. 13 Camera-scene distances on site used to run the AR app.

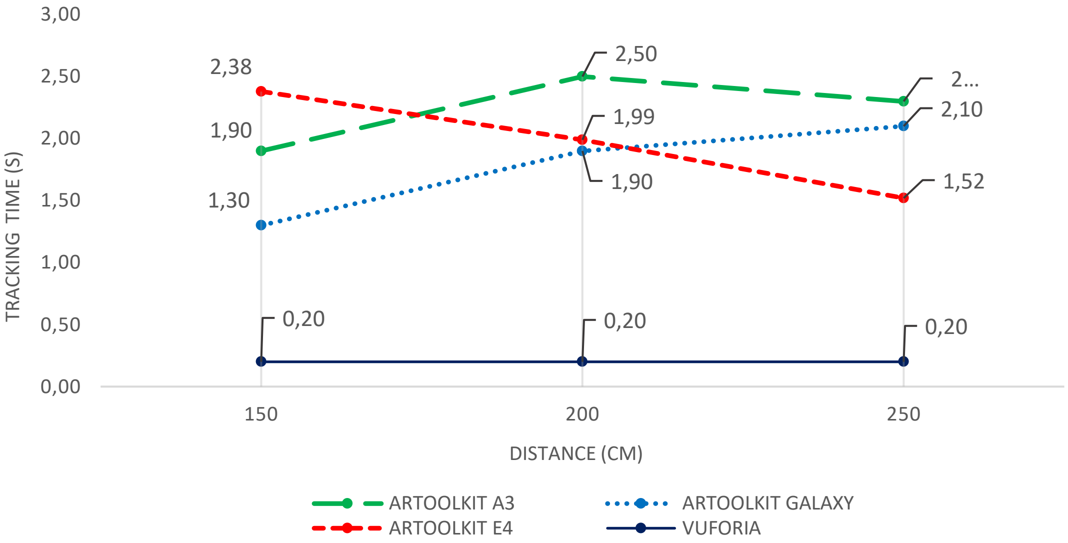

Fig. 14 Tracking times obtained with the three smartphones at different distances; no differences were found among smartphones with Vuforia.

5 Discussion

This research evaluates a list of factors that affect the user experience in an AR app for disseminating rock art paintings. While other studies are focused on marker-based tracking assessment (listed in Section 1), the present study is aimed at environments where the use of markers is not allowed at all.

Some studies analyse the effect of marker-camera distances. Dos Santos et al. [17] analysed the distance between marker and camera and conclude that the minimum and maximum distance reached with Vuforia is better than ARToolKit. In the case of image tracking, there are not relevant differences between the evaluated libraries; the maximum camera-scene distance is shorter than 10 cm. Rabbi et al. [18] proved that the error increases as marker distance is longer, causing marker tracking failure. In the image tracking studied herein, the tracking time remains more or less constant with increasing distance, thus the user can be placed at different distances from the paintings.

The tracking time with Vuforia was minimum, less than 1/2 s, regardless of the camera-scene distance. ARToolKit required more time to obtain the camera pose and load virtual content, around 2 or 3 s, depending on the smartphone used. These longer times may be owing to the great amount of features detected by ARToolKit. The feature database was calculated using the highest accuracy, obtaining the maximum number of possible features. Hence, the ARToolKit database was definitely much larger than Vuforia's. This meant that the app had to recognise more points in real time to calculate the camera pose, increasing the tracking time. Surely with less points (smaller datasets), ARToolKit would work faster. In addition, the app initialisation time in ARToolKit lasted twice than Vuforia, about 15 s to be ready to be used.

Some authors studied common problems in marker-based tracking, such as marker occlusions [45, 46]. As demonstrated in this study, the image occlusion does not affect the markerless tracking; tracking time remains unchanged with different occlusion percentages. This result is achieved because markerless tracking makes use of a great number of features well-distributed throughout the image. Nevertheless, this statement may vary depending on the number of features covered in the targeting area.

On the other hand, extreme changes in lighting conditions can affect the detection of features. According to Khan et al. [16], higher light intensity in a shadow environment provided faster marker detection while direct sunlight over yielded greater errors. This also happened in our study, where the experiments demonstrated that the ARToolKit's tracking time increases when the image exposure changes. However, different exposures did not affect the Vuforia's tracking system.

The tracking system requires a previous image feature extraction to generate the features database. ARToolKit detected a large amount of features in all the images that were tested. Thus, the scene identification and tracking was possible with both scenarios, indoor using a printed colour image and outdoor on site in the real shelter. Instead, Vuforia did not always detect enough features. The printed colour images presented in Fig. 4a for the indoor assessment had to be enhanced to allow the software to detect enough features and thus to make possible the scene recognition and tracking (Fig. 4b). Others scenes were tested in Cova dels Cavalls (Fig. 12), but Vuforia only worked in the scene studied. Therefore, it can be stated that it is not recommended to develop with Vuforia an AR rock art app containing all the paintings presented in Cova dels Cavalls as a case study. However, ARToolKit was ready to find and match many features in all scenes.

The proper positioning of a 2D image into the 3D world depends on the camera calibration. If the intrinsic and extrinsic camera parameters were unknown, virtual content would not be placed in the exact 3D position. ARToolkit provides a server with calibration parameters for some smartphones, but it is advisable to calibrate the camera to achieve better results. In the cases analysed, the virtual content was not correctly placed on top of the screen without camera calibration, hence the three smartphones had to be calibrated. Conversely, the virtual content was accurately positioned without prior calibration using Vuforia library.

On the other hand, ARToolKit library did not include camera autofocus, hence the image were usually blurred. In addition, without autofocus, image recognition did not work at medium range. This was an ARToolKit fault, but it can be solved by modifying the camera script.

Regarding image flickering, after testing both libraries, flickering and blurring of the virtual image was practically non-existent on Vuforia. However, ARToolKit results were a bit worse.

Finally, ARToolKit is one of the most important open source AR library available nowadays; this is a great advantage because the whole code is completely open. Thus, the programmer can modify and adapt it to his/her needs. In ARToolKit, for instance, the programmer is free to modify the tracking algorithm. In addition, currently new improvements will be hopefully released in version 6. However, this freedom is not expected in Vuforia, where it is not possible to access the AR code and modify it.

Overall, Table 1 summarises the main results achieved after this user and provider (developer) experience assessing our developed AR app for disseminating rock art with both libraries, ARToolkit and Vuforia.

Table 1 Main results from the ARToolkit and Vuforia comparison.

Features	**Vuforia**	**ARToolKit**
App initialisation time	Around 7.5 s	Around 14.5 s
Image occlusion	Does not affect	Does not affect
Tracking time	0.35 s	4.5s – 1.2s
Lighting conditions	Does not affect	Tracking time increases with the exposure
Poor detail reference image (feature database)	Low number of keypoints	High number of keypoints
Required camera calibration	No	Yes
Camera autofocus	Yes	No
Flickering and blurring	No	Low
Open source / the code can be modified	No	Yes

6 Conclusion

In this paper, we have presented a comparative study of ARToolKit and Vuforia libraries applied to the recognition of natural features in rock art scenarios. Several experiments have been performed in both indoor and outdoor environments to analyse the factors that affect the user experience. Thus, it is easier to decide from a provider (developer) point of view which of the two libraries is the best to continue developing AR apps to disseminate rock art paintings.

Taking into account the assessment results summarised in Table1, Vuforia provided a better user experience and developers will also be able to design AR apps more quickly. However, Vuforia's feature extraction did not work well in images containing poor details, as it is the case of rock art paintings. For complex outdoor environments such as the ones presented in prehistoric rock art sites with paintings, the AR apps cannot be developed with Vuforia, however ARToolKit library recognised many features in each of the Cova dels Cavalls scenes, hence it is recommended to apply ARToolKit to develop

AR apps in outdoor rock art environments. In addition, as ARToolKit is an open source library, there is no limitation (far beyond its maintenance in the future) because the source code can be modified. Moreover, its documentation is comprehensive and a good user forum exists at present.

In the future, we plan to develop new AR apps related to Levantine prehistoric rock art motifs using the ARToolKit library. To reduce the unfavourable tracking time on some devices using ARToolKit, additional research will be devoted to determine the right number of features (threshold) able to guarantee a correct pose estimation and subsequent image reprojection.

Acknowledgement The authors gratefully acknowledge the support from the Spanish Ministerio de Economía y Competitividad to the project HAR2014-59873-R. Similarly, the authors want to express their gratitude to the General Directorate of Culture and Heritage, Conselleria d'Educació, Investigació, Cultura i Esport, Generalitat Valenciana for letting us access and carry out research at the archaeological site.

References

1. Siltanen S (2015) Diminished reality for augmented reality interior design. Vis Comput 33:1-16 . doi: 10.1007/s00371-015-1174-z
2. Kim SL, Suk HJ, Kang JH, et al (2014) Using Unity 3D to facilitate mobile augmented reality game development. Internet Things (WF-IoT), 2014 IEEE World Forum 21-26 . doi: 10.1109/WF-IoT.2014.6803110
3. Carmigniani J, Furht B, Anisetti M, et al (2011) Augmented reality technologies, systems and applications. Multimed Tools Appl 51:341-377 . doi: 10.1007/s11042-010-0660-6
4. Azuma R (1997) A survey of augmented reality. Presence Teleoperators Virtual Environ 6:355-385 . doi: 10.1.1.30.4999
5. Amin D, Govilkar S (2015) Comparative Study of Augmented Reality Sdk'S. Int J Comput Sci Appl 5:11-26 . doi: 10.1227/01.NEU.0000297044.82035.57
6. Azuma R, Baillot Y, Feiner S, et al (2001) Recent Advances in Augmented Reality. Ieee Comput Graph Appl 34-47 . doi: 10.4061/2011/908468
7. Blanco-Pons S, Carrión-Ruiz B, Lerma JL (2016) REVIEW OF AUGMENTED REALITY AND VIRTUAL REALITY TECHNIQUES IN ROCK ART. En: Proceedings of the 8th International Congress on Archaeology, Computer Graphics, Cultural Heritage and Innovation 'ARQUEOLÓGICA 2.0'. pp 176-183
8. Brancati N, Caggianese G, Frucci M, et al (2017) Experiencing touchless interaction with augmented content on wearable head-mounted displays in cultural heritage applications. Pers Ubiquitous Comput 21:203-217 . doi: 10.1007/s00779-016-0987-8
9. Gutierrez JM, Molinero MA, Soto-Martín O, Medina CR (2015) Augmented Reality Technology Spreads Information about Historical Graffiti in Temple of Debod. Procedia Comput Sci 75:390-397 . doi: http://dx.doi.org/10.1016/j.procs.2015.12.262
10. Ridel B, Reuter P, Laviole J, et al (2014) The Revealing Flashlight: Interactive Spatial Augmented Reality for Detail Exploration of Cultural Heritage Artifacts. J Comput Cult Herit 7:6:1-6:18 . doi: 10.1145/2611376

11. Haladová ZB, Szemzö R, Kovačovský T, Žižka J (2015) Utilizing Multispectral Scanning and Augmented Reality for Enhancement and Visualization of the Wooden Sculpture Restoration Process. Procedia Comput Sci 67:340-347 . doi: 10.1016/j.procs.2015.09.278

12. Pierdicca R, Frontoni E, Zingaretti P, et al (2015) Making Visible the Invisible. Augmented Reality Visualization for 3D Reconstructions of Archaeological Sites. Augment virtual Real Second Int Conf AVR 2015 9254:25-37 . doi: 10.1007/978-3-319-22888-4

13. Chen CY, Chang BR, Huang P Sen (2014) Multimedia augmented reality information system for museum guidance. Pers Ubiquitous Comput 18:315-322 . doi: 10.1007/s00779-013-0647-1

14. Domingo I, Carrión B, Blanco S, Lerma JL (2015) Evaluating conventional and advanced visible image enhancement solutions to produce digital tracings at el Carche rock art shelter. Digit Appl Archaeol Cult Herit 2:79-88 . doi: http://dx.doi.org/10.1016/j.daach.2015.01.001

15. Carrión-Ruiz B, Blanco-Pons S, Lerma JL (2016) DIGITAL IMAGE ANALYSIS OF THE VISIBLE REGION THROUGH SIMULATION OF ROCK ART PAINTINGS. En: Proceedings of the 8th International Congress on Archaeology, Computer Graphics, Cultural Heritage and Innovation 'ARQUEOLÓGICA 2.0'. pp 169-175

16. Khan D, Ullah S, Rabbi I (2015) Factors affecting the design and tracking of ARToolKit markers. Comput Stand Interfaces 41:56-66 . doi: 10.1016/j.csi.2015.02.006

17. Dos Santos AB, Dourado JB, Bezerra A (2016) ARToolkit and Qualcomm Vuforia: An Analytical Collation. Proc - 18th Symp Virtual Augment Reality, SVR 2016 229-233 . doi: 10.1109/SVR.2016.46

18. Rabbi I, Ullah S, Javed M, Zen K (2014) Analysis of ARToolKit Fiducial Markers Attributes for Robust Tracking. 1st Int Conf Recent Trends Inf Commun Technol Anal 281-290

19. Kounavis CD, Kasimati AE, Zamani ED (2012) Enhancing the tourism experience through mobile augmented reality: Challenges and prospects. Int J Eng Bus Manag 4:1-6 . doi: 10.5772/51644

20. ARToolkit (2017) ARToolkit. https://archive.artoolkit.org/. Accessed 2 oct 2017

21. Vuforia (2017) Vuforia. https://www.vuforia.com/. Accessed 2 oct 2017

22. Fischer J, Eichler M, Bartz D, Straßer W (2007) A hybrid tracking method for surgical augmented reality. Comput Graph 31:39-52 . doi: 10.1016/j.cag.2006.09.007

23. Sörös G, Seichter H, Rautek P, Gröller E (2011) Augmented visualization with natural feature tracking. Proc 10th Int Conf Mob Ubiquitous Multimed 4-12 . doi: 10.1145/2107596.2107597

24. González C, Vallejo D, Albusac J, Castro J (2011) Realidad Aumentada. Un enfoque práctico con ARToolKit y Blender. 2-120

25. Vuforia (2017) Vuforia-VuMark. https://library.vuforia.com/articles/Training/VuMark. Accessed 4 abr 2017

26. Wang G, Wang B, Zhong F, et al (2015) Global optimal searching for textureless 3D object tracking. Vis Comput 31:979-988 . doi: 10.1007/s00371-015-1098-7

27. Radkowski R, Oliver J (2013) Natural feature tracking augmented reality for on-site assembly assistance systems. En: Shumaker R (ed) Virtual, Augmented and

Mixed Reality. Systems and Applications. VAMR 2013. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 281-290

28. Uchiyama H, Marchand E (2012) Object Detection and Pose Tracking for Augmented Reality: Recent Approaches. 18th Korea-Japan Jt Work Front Comput Vis 1-8

29. Marchand E, Uchiyama H, Spindler F, et al (2016) Pose estimation for augmented reality : a hands-on survey. IEEE Trans Vis Comput Graph 22:2633-2651 . doi: 10.1109/TVCG.2015.2513408

30. Moreels P, Perona P (2007) Evaluation of Feature Detectors and Descriptors based on 3D Objects. Int J Comput Vis 73:263-284 . doi: 10.1007/s11263-006-9967-1

31. Lowe DG (2004) Distinctive image features from scale invariant keypoints. Int J Comput Vis 60:91-11020042 . doi: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

32. Wu S, Oerlemans A, Bakker EM, Lew MS (2017) A comprehensive evaluation of local detectors and descriptors. Signal Process Image Commun 59:150-167 . doi: 10.1016/J.IMAGE.2017.06.010

33. Alahi A., Ortiz R., Vandergheynst P (2012) FREAK: Fast Retina Keypoint. Comput Vis Pattern Recognit 510-517 . doi: 10.1109/CVPR.2012.6247715

34. Unity Unity. https://unity3d.com/es. Accessed 12 oct 2017

35. Engine U (2017) Unreal Engine. https://www.unrealengine.com/. Accessed 10 oct 2017

36. CRYENGINE CRYENGINE | The complete solution for next generation game development by Crytek. https://www.cryengine.com/. Accessed 7 jun 2017

37. Fiala M (2005) ARTag, a fiducial marker system using digital techniques. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2:590-596 . doi: 10.1109/CVPR.2005.74

38. ArUco ArUco: a minimal library for Augmented Reality applications based on OpenCV | Aplicaciones de la Visión Artificial. https://www.uco.es/investiga/grupos/ava/node/26. Accessed 19 abr 2018

39. DroidAR (2017) DroidAR by bitstars. https://bitstars.github.io/droidar/. Accessed 10 dic 2017

40. Vuforia (2017) Image Targets. https://library.vuforia.com/articles/Training/Image-Target-Guide. Accessed 11 abr 2017

41. ARToolkit (2017) About. https://artoolkit.org/about-artoolkit. Accessed 11 abr 2017

42. ARToolkit (2017) Documentation. https://artoolkit.org/documentation/. Accessed 12 abr 2017

43. Camera-Calibration Camera Calibration App for Android [ARToolkit]. https://archive.artoolkit.org/documentation/doku.php?id=4_Android:android_camera_calibration. Accessed 16 oct 2017

44. Martínez R, Villaverde V (2002) La cova dels cavalls en el Barranc de la Valltorta

45. Liu S, Ge S, Yu H (2016) Research on Robustness Recognition Algorithms in Augmented Reality. 3rd Int Conf Inf Sci Control Eng 547-552 . doi: 10.1109/ICISCE.2016.123

46. Seo J, Shim J, Choi JH, et al (2011) Enhancing marker-based AR technology. En:

International Conference on Virtual and Mixed Reality. Virtual and Mixed Reality - New Trends. Springer, Berlin, Heidelberg, pp 97-104