EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2018)
R. Sablatnig and M. Wimmer (Editors)

Evaluation of Multi-Platform Mobile AR Frameworks for Roman Mosaic Augmentation

Jorge C. S. Cardoso and André Belo

CISUC - Department of Informatics Engineering, University of Coimbra, Portugal

Abstract

Augmented Reality (AR) development frameworks have different recognition performance on different kinds of target images. In this work, we studied AR frameworks applied in the context of outdoor Roman mosaic ruins with the final aim of developing a multi-platform mobile AR application. We started by analysing the documented features of existing frameworks to determine the feasible ones. This resulted in the selection of three frameworks: CraftAR, PixLive, and Wikitude. We then experimentally evaluated the performance of the target recognition features against real mosaics by measuring the recognition delay, minimum required target area, visual alignment and visual stability. Results indicate a good recognition rate for CraftAR and a poor recognition rate for Wikitude. CraftAR showed better recognition delay and visual stability, while PixLive showed better minimum required target area and visual alignment.

CCS Concepts

•***Human-centered computing*** → ***Mixed / augmented reality; Empirical studies in ubiquitous and mobile computing;*** *Empirical studies in HCI; Smartphones;* •***Computer graphics*** → ***Mixed / augmented reality;***

1. Introduction

AR has been used extensively in Virtual Heritage (VH) contexts as an aid for visualizing, understanding, teaching, and experiencing ancient artefacts and cultures.

> "*In virtual heritage, AR is used to enhance the overall experience of the visitor of a cultural heritage site. . . . it can enhance, motivate and stimulate students' understanding of certain events. . .*" [NSP09]

Systems that take advantage of different types of devices for experiencing AR and with different VH goals have been built, evaluated, and put to public use. In the Archeoguide, for example, Vlahakis et al. [VIK*02] used a portable head-mounted display with variable transparency to create an AR system that allowed visitors of ancient Greek ruins to experience architectural reconstruction of the temple of Hera. In the PRISMA project, Fritz et al. [FSL05] describe an augmented binoculars system where tourists can see the view from one of the hills of the city of Donostia-San Sebastian, enhanced with digital information. In LecceAR, Scopino et al. [SGM*15] describe a smartphone AR application that recognizes a printed image of the Roman Amphitheatre in Lecce and overlays a three-dimensional architectural reconstruction.

In this work, we are concerned with the application of AR to the Roman mosaics [con17b] of the Conímbriga archaeological site. Conímbriga is one of the largest Roman settlements excavated in Portugal, located in the municipality of Condeixa-a-Nova, 16 kilometres from Coimbra [con17a]. Excavations have revealed several structures and buildings with Roman mosaic floors. The House of Fountains (Figure 1) is one such building – a house with a water garden and peristyle.

Figure 1: *View of the house of fountains and the mosaic in one of the rooms.*

Our aim is to create a mobile AR system that can:

- Provide additional textual information about the mosaics, for example, when they were uncovered, what was the latest conservation or restoration work, etc.
- Provide additional conservation and restoration information, for example, display image overlays of the conservation or restoration works on mosaics over time.
- Provide a platform for the visualization of virtual restoration of the existing mosaics. The virtual restoration images could be created by different audiences with different purposes. For example, in the context of a school visit, students could digitally manipulate mosaic images and creatively "restore" missing parts which could then be experienced through the AR application.
- Provide visual indication regarding specific aspects of the various mosaics. For example, mosaics could be highlighted with graphical information regarding the geometric patterns employed.

These aims result in the following requirements for an AR framework:

1. Given the usage context of our application, the AR framework needs to support natural image feature recognition (i.e., capable of recognizing natural images – the real mosaics – instead of requiring specially crafted visual markers).
2. Our application needs to be able to overlay, at least, 2D images over the detected mosaics (and possibly also video, 3D objects, text, and audio).
3. The application needs to be informed about the identification of recognized mosaics, so that further information may be displayed to the user outside the AR view.
4. The mobile AR application should be developed in a multi-platform framework so as to minimize development and maintenance costs. Although we are interested only on mobile applications, mobile platforms (e.g. Android, iOS, Windows) are very distinct and usually require different application code bases. A multi-platform development framework allows developers to reuse the same code base and target multiple execution platforms (i.e., the same code can be used to generate Android, iOS, and Windows applications).

The applicability of existing AR frameworks in this context, however, is not obvious. Most AR frameworks with natural image feature detection have been optimized for printed images. Furthermore, Conímbriga is an outdoor, and mostly unprotected, site. This may impose additional difficulties for AR frameworks: lighting is uncontrollable and may vary much due to weather conditions which may impact the brightness and contrast of the mosaics as captured by the smart device; although mosaics are subject to cleaning and maintenance, dust, weeds, and fungi may appear resulting in mosaics images that vary from the reference image used by the AR framework.

Using an existing AR framework however, represents less development effort than creating a custom image recognition technique. Given the requirements and context of use for this AR applications, our first approach is thus to understand which existing AR development frameworks are suitable, and compare their image recognition performance on the specific use case of real Roman mosaics.

The contributions of this work are thus:

- An analysis of the features of existing AR frameworks regarding the requirement for natural image features recognition and support for multi-platform development. This analysis can be used by other Cultural Heritage professionals with similar requirements for AR applications, even if in domains other than Roman mosaics.
- A comparison of the performance of the natural image features recognition when applied to outdoor mosaics. To our knowledge, this is the first study to explicitly compare the performance of AR frameworks on Roman mosaic images. Although we focused on Roman mosaics, our results show highlight the fact that different AR frameworks may have very different behaviour on the same image type.

2. Related Work

With a focus on the development of educational AR applications, Herpich et al. [HGT17] analysed eleven AR frameworks regarding the features they offered to application developers. The analysed frameworks where: ARToolKit, Augment, Aurasma, BlippAR, CraftAR, EasyAR, Kudan, LayAR, PixLive, Vuforia, and Wikitude. The authors evaluated each framework according to whether it provided a given feature. Features were classified into "target tracking types", and "additional features" categories.

The "target tracking types" category included:

Text The framework offers the capability of recognizing text (e.g. Optical Character Recognition (OCR)).
Planar Image The framework offers the capability of recognizing planar images.
3D Object The framework offers the capability of recognizing pre-defined objects (cylindrical, conical) or general 3D objects.
Multi Targets The framework is capable of detecting multiple targets simultaneously.
Geo-location The framework supports geolocated targets (e.g., Point Of Interest (POI)).
Markerless The framework supports any markerless detection technology (e.g. Simultaneous Localization and Mapping (SLAM)).

The "additional features" category included:

Online recognition The framework allows tracking targets available through an online service.
Offline recognition The framework allows tracking targets available offline, without requiring any connection to the Internet.
AR Editing Platform The framework provides a web-based platform for creating and managing targets and building and editing resources.

Each feature present in the framework was scored with 1 point, except for the "AR Editing Platform" which was scored with 2 points. Results showed that Wikitude and Vuforia scored the highest number of points (9 and 8). The authors also list, for each framework, the supported development platforms (e.g., Windows, Mac OS, Linux, Android, iOS) and available extensions (e.g., availability of Unity packages). This study provides a good overview of AR frameworks and is useful as a starting point. However, the analysis provides only a high-level view of the features of the various

frameworks, lacking fundamental details needed for specific applications. For example, the "planar image" tracking type feature does not distinguish between frameworks that are only capable of detecting fiducial markers from those that support training arbitrary images as markers (i.e., through natural features detection). In our work, we are concerned with frameworks that can use arbitrary images because placing artificial fiducial markers near the Roman mosaics in the ruins is currently not an option.

Amin and Govilkar [AG15] analysed 6 AR frameworks: Metaio (which has since been acquired by Apple and discontinued), Vuforia, Wikitude, D'Fusion, ARToolkit, and ARmedia. These frameworks are compared on several categories such as licensing type (open source, free, commercial), supported development platform (iOS, Android, Windows, Linux, Flash, BlackBerry), availability of marker generation tools (e.g., online marker management tool, pre-defined marker dataset), type of tracking (marker, Global Positioning System (GPS), Inertial Measurement Unit (IMU), face tracking, natural image features, 3D object, other), type of overlay media (2D content, 3D content, other – animation, HyperText Markup Language (HTML) content). The type of tracking comparison provided by this study explicitly details the capabilities of the frameworks in terms of natural features tracking and concludes that all analysed frameworks provide natural image features tracking. However, this and the study by Herpich et al. [HGT17] provide only an analysis on the presence or absence of features of the framework. In our current study, we are interested in understanding also the performance of the natural image features tracking when applied to a very specific type of scene/image – real Roman mosaics.

Marneanu et al. [MER14] focused on AR mobile application development for Android devices and evaluated six frameworks – ARLab, ARToolkit, D'Fusion, Vuforia, Catchoom, and Metaio (although no references are provided for the analysed frameworks, we assume that Catchoom refers to CraftAR [Cat18]). Contrary to the previously described studies, Marneanu et al. [MER14] evaluated the performance of the chosen frameworks. They developed a test mobile application and ran the various frameworks through it in order to measure the tracking performance against various conditions in two main categories: environmental, and target.

Environmental conditions included:

Light intensities Varying lighting conditions (normal lighting, overexposing with desk lamp, light changes, mirroring effect)
Background Varying the background (light or dark) on which the target rests.
Viewpoint Varying viewpoint in four different perspectives: 45^o angles to the left and right, upwards and downwards.
Visibility Varying visibility by uncovering 10% of the target at each step until reaching 100% visibility.
Noise Varying digitally introduced noise into the target images.
Distance Varying the distance to the target image starting at 10cm with 10cm increments until the target can no longer be recognized.

Target conditions included:

Grayscale Printed default size target image in grayscale.
Contrast Four digitally modified target images starting with a contrast value set to -50, incremented by 50 at each step.
Size Four target image sizes of 5cm, 10cm, 15cm and 20cm while keeping the distance of the testing device to the target at 30cm.
Aspect ratio Digitally modified target images, shrunk either vertically or horizontally by reducing the height or the width to one third (1/3), a half (1/2) and two thirds (2/3).
Material Simulation of target image in different materials simulating recurrent situations: behind glass (e.g., poster inside panels on bus shelter), laminated (e.g., restaurant menu), printed on glossy photo paper (e.g., promotional ad).

Marneanu et al. [MER14] benchmarked each AR framework in the various conditions by measuring how long it took to detect the target or at what distance the framework was unable to detect the target image. They also defined several scenarios and analysed which framework would be best for a given scenario given the benchmarking results. In the outdoors scenario – a bus shelter situation where the user is inside the bus – they conclude that Vuforia would be the best choice. Marneanu et al. [MER14] provide a very complete test suit for AR frameworks but they focused only on the ones supporting Android development. Also, all testing conditions make use of printed target images. In our work, we are interested in finding how AR frameworks behave when detecting real stone Roman mosaics, not printed images of mosaics.

Rautenbach et al. [RCJ16] analysed AR frameworks suitable for the development of an AR application capable of displaying house addresses in a live outdoor view through the mobile device. The authors used the online comparison table of AR frameworks on SocialCompare – an online collaborative comparison service – as the basis for their search. To evaluate the frameworks, they defined a set of criteria in three categories: general criteria (platform, programming language, license, use of standards), function criteria (offline availability, data source, data display, object behaviour, display radius, visual search), and non-functional criteria (ease of integration with other applications, ease of extending the framework, usability, documentation). The "visual search" criteria refers to the frameworks capability of recognizing specific objects based on, e.g., a photo. As in other similar studies, the analysis by Rautenbach et al. [RCJ16] focuses on the presence of absence of features on the framework. They did not explicitly compare the performance of the tracking features of the analysed frameworks.

3. AR Framework Selection Study

In order to find and filter AR frameworks suitable for multi-platform mobile application development we started by looking at existing studies and web sites. In her master dissertation, Marto [Mar17] analysed and compared 48 AR frameworks providing a comparison table that includes licensing type, mobile platform compatibility, compatibility with Unity 3D, GPS and IMU support, and tracking type. The website SocialCompare provides a similar comparison table [Soc18] made collaborative by users of the website. These two resources, along with the related works described earlier, and each framework's documentation website, provided us with the main input for filtering frameworks that offered both multi-platform development and natural image features tracking.

We started by creating a list of AR frameworks [BC18a] that supported mobile application development. We focused on support for multi-platform mobile development and on the tracking features offered by the framework.

Analysis of the framework list, reveals 6 frameworks with official multi-platform mobile development support: Augment [Aug18], CraftAR [Cat18], ezAR [eT18], Layar [Lay18a], PixLive [Vid18], and Wikitude [Wik18].

We then analysed each of these 6 frameworks more closely by reading their documentation, creating an account to test the tracker upload process and, in some cases, implementing test applications. We wanted to better understand the application development process for each of these frameworks and learn about any limitations. Table 1 shows the main characteristics of the AR frameworks. The multi-platform column indicates AR frameworks that have official support for multi-platform development (we found some unofficial extensions for some of the frameworks but these do not provide any guarantees of continuing support and so were not considered).

Augment is focused on a shopping experience for 3-dimensional product visualization. They provide a mobile application which can be used in markerless or marker-based AR to visualize products uploaded through their website. Products are uploaded as 3D models only, there is no support for 2D media such as images or videos. Augment provides an Software Development Kit (SDK) and a Cordova plugin, however, analysis of the example application built with the Augment SDK reveals that the it provides only a markerless AR experience – overlaying a pre-chosen model over the environment. This makes the Augment framework unsuitable for the purposes of this project.

ezAR provides only face detection and geo-location AR capabilities, which makes it unsuitable given the requirements for our mosaic application. Additionally, during the course of this study, ezAR seems to have been discontinued as their website now shows a domain purchase notice. Hence, this framework was excluded from further analysis.

Layar provides an AR service where customers can create AR campaings by uploading image targets and associating overlay media content and interactive components. A campaign can then be published and experienced through the Layar mobile application. Layar provides an SDK for custom application development that includes natural image feature detection along with geo-localized targets. However, the Cordova/PhoneGap plugin does not support callbacks to inform the application about which targets have been recognized: "*Layer SDK callbacks are not supported in the plugin. Using the PhoneGap plugin you can launch a scan view or open a layar but cannot get javascript callbacks about what is going on.*" [Lay18b].

For our AR mosaic application, feasible frameworks are thus: CraftAR, PixLive, and Wikitude.

4. Experiment

With this experiment, we wanted to understand how each of the three AR frameworks performed with real mosaic images. We did this by:

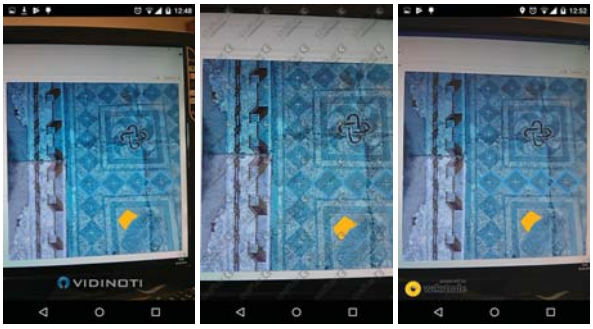
1. Creating 3 mobile applications, each one with a different AR framework (application development).
2. Configuring each application with the same set of mosaic targets, collected from the Conímbriga site, and overlay images (target acquisition).
3. Video recording each application in real-context usage at the Conímbriga site by pointing the camera to the selected mosaics and performing the same set of gestures with all three applications (video recording).
4. Analysing the videos to determine differences in the recognition delay, minimum required target area for tracking, visual alignment and visual stability (video analysis).

Each of these steps is described next in more detail.

4.1. Application development

All three applications were developed using the Cordova mobile application development framework, following the guidelines for each of the AR frameworks for inclusion and usage of their Cordova plugin. All applications were created based on the most simple example provided by each of the AR frameworks. Applications consisted of a single screen (AR view) on which overlay images were superimposed when the target was recognized (see Figure 2).

We used the AR framework demonstration license on all applications. CraftAR and Wikitude applications used offline tracking, PixLive used online tracking but overlays were pre-downloaded.



(a) *PixLive* **(b)** *CraftAR* **(c)** *Wikitude*

Figure 2: *Screen captures of the three applications recognizing an image of mosaic "Mosaico1_c" on the computer screen.*

4.2. Target Acquisition

Target acquisition was a difficult process because most mosaics are off-limits for visitors and many can only be seen from a distance. Some mosaics are surrounded by waist-height walls, others are visible only from artificially built elevated passage ways.

To capture the target images, we photographed the mosaics from the position a regular visitor would have. We tried to take photographs as parallel to the floor as possible, in some cases using a selfie-stick to help level the camera (see Figure Figure 3). We used a Samsung Galaxy J5 phone and a Canon EOS 1200D to take the photographs. Photographs were taken in a sunny day in the afternoon.

Nome	Multi-platform mobile development	Mobile platform	Tracking								Overlay types
			2D image tracking (natural features detection)	3D object tracking	Marker based tracking (Fiducial)	Markerless tracking (SLAM)	Geo location	Offline tracking	Online tracking	Multiple targets	
Augment	Y	Android, iOS	Y	N	Y	Y	N	Y	Y	N	3D objects only
CraftAR	Y	Android, iOS	Y	N	N	N	N	Y	Y	Y	Image, Video, 3D Model
EzAR	Y	Exclusivo para Cordova	n.e.	Y (face detection)	N	N	Y	n.e.	n.e.	n.e.	n.e.
Layar	Y	Android, iOS, BlackBerry	Y	N	Y	n.e.	Y	N	Y	N	Video, Image, Sound, 3D Model
PixLive	Y	Android, iOS	Y	N	n.e	n.e.	Y	Y	Y	N	Image, Video, Audio, 3D Model, Text
Wikitude	Y	Android, iOS, Windows Phone, BlackBerry	Y	Y	Y	Y	Y	Y	Y	Y	Image, Video, 3D Object, Text

Table 1: *Main features of the analysed AR frameworks. Highlighted rows represent the frameworks chosen for the experiment.*

Figure 3: *Target acquisition.*

The photographs were then edited to remove the parts of the image that did not belong to the mosaic. From the set of photographed mosaics, we chose 20 to be part of the experiment. Mosaics images were chosen to include mosaics with different patterns and figures, in different conservation state, and photographed in different lighting conditions. We purposefully kept some images with shadows cast upon the mosaics, although these do not seem *a priori* good target images, to determine the AR frameworks would perform on such targets. The complete set of target images used in this work can be seen in Figure 4 (full images can be retrieved from [BC18d]).

We then configured the targets using the framework's web interface to generate the trackers' files (all three chosen frameworks provide a web interface). CraftAR and Wikitude provide target image ratings after they are uploaded – an indication of how well the tracker will work on those images. For reference, the target image ratings are listed in Table 2. In general, the target images were well rated by the two frameworks – only two images received rating of 1/3 (one star out of three possible) by Wikitude, and only one image received a rating of 3/5 by CraftAR, with all other images rated higher. Fifty percent of the target images received 3/3 by Wikitude, and eighty-five percent received a 5/5 by CraftAR. The algorithms used to calculate these ratings are not disclosed, however.

Table 2: *Target ratings as indicated by the AR framework interface after target image processing. PixLive does not provide a target rating.*

Mosaic	CraftAR	Wikitude
Mosaico 1_c	5/5	2/3
Mosaico 3_e	5/5	2/3
Mosaico 6_d	3/5	1/3
Mosaico 9_f	4/5	2/3
Mosaico 10_b	5/5	2/3
Mosaico 13_f	5/5	2/3
Mosaico 14_g	5/5	2/3
Mosaico 15_f	4/5	1/3
Mosaico 19_a	5/5	3/3
Mosaico 20_b	5/5	3/3
Mosaico 21_a	5/5	3/3
Mosaico 22_e	5/5	3/3
Mosaico 23_b	5/5	3/3
Mosaico 24_d	5/5	3/3
Mosaico 25_c	5/5	3/3
Mosaico 26_d	5/5	3/3
Mosaico 27_a	5/5	2/3
Mosaico 28_d	5/5	3/3
Mosaico 29_b	5/5	3/3
Mosaico 31_a	5/5	2/3

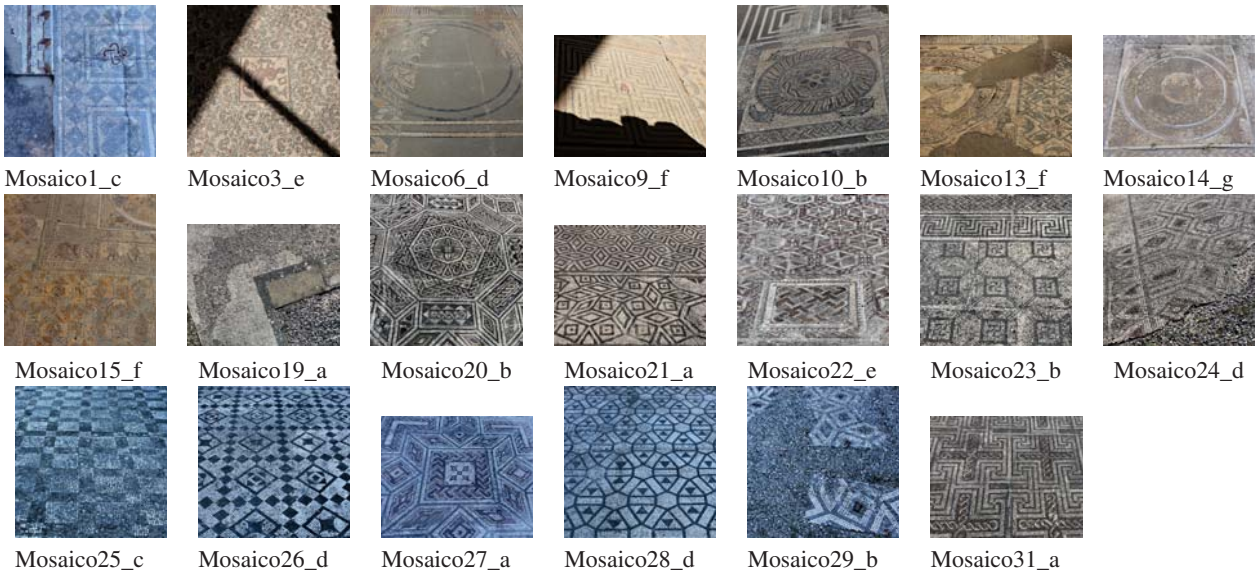


Figure 4: *Targets*

We then created overlay images, using the target images as reference, by marking some contours of the mosaics and adding a few filled shapes. The purpose of these overlay images was simply to allow an easy visual inspection of the resulting alignment by the AR framework – by analysing how close the contours and shapes match the mosaic image. An example target and overlay image is provided in Figure 5 (the full set of overlay images can be retrieved from [BC18c]).

(a) *Target* **(b)** *Overlay* **(c)** *Overlay on target*

Figure 5: *Example of target and overlay images.*

4.3. Video recording

After creating the image targets and developing each application, we went back to Conímbriga and performed the experiment session. The session happened in a day with similar weather conditions as the day the target photos were originally taken.

For each mosaic in our target set, we recorded the operation of each of the three applications/frameworks applying three types of movements:

1. Starting with the smartphone pointing down, we lifted it to a stable position – the point where the mosaic was in view – and waited for the image overlay to appear. We then moved the smartphone to the left until the target was out of view and then to the right (see Figure 6a).
2. Starting with the smartphone pointed at the mosaic and visible at 100%, we slowly moved it to the left and right and to the top and bottom. Motions were made until about 30% of the target was visible (see Figure 6b).
3. Starting with the smartphone pointed at the mosaic and visible at 100%, we slowly moved towards the mosaic until our arm was fully extended, and then backwards (see Figure 6c).

We performed the recordings by hand (i.e., without the use of any tripod or any motorised arm) to emulate a natural use of the application.

The test applications ran in a Redmi Note 4 smartphone and videos were recorded using the built-in ScreenRecorder application. For each mosaic, AR framework, and movement, we recorded one video. Videos were coded with the number of the movement they referred to.

4.4. Video analysis

After the experiment session we analysed the recorded videos to compare the AR frameworks. We measured the following parameters: recognition delay, minimum required target area, visual alignment, visual stability.

4.4.1. Recognition delay

Recognition delay measures how long the framework needs to recognize the target and display the overlay image.

To measure recognition delay we analysed video from movement #1, using as reference the first frame of the stable position of the smartphone with the target visible as indicated by Figure 6a2. From this frame we counted forward the number of frames (positive number) until the overlay image first appears, or we counted backwards the number of frames (negative number) until the overlay image disappeared. In the cases where the application detects

(a1) *Pointing down* **(a2)** *Target and overlay in view* **(a3)** *Movement left* **(a4)** *Movement right*

(a) *Movement #1*

(b1) *Target and overlay in view* **(b2)** *Movement left* **(b3)** *Movement right* **(b4)** *Movement top* **(b5)** *Movement bottom*

(b) *Movement #2*

(c1) *Target and overlay in view* **(c2)** *Movement towards target* **(c3)** *Movement away from target*

(c) *Movement #3*

Figure 6: *Movements – Example screenshots.*

the target even before the stable smartphone position is reached, this is counted as a negative delay. We chose the stable position of the smartphone as the reference because the way videos were recorded in the field, we could not guarantee that all the videos had the exact same duration and that the target was visible at the exact same time in all videos.

After counting the number of frames, we converted the number to seconds using the video's framerate (each video had a slightly different framerate, so we used each video's encoded framerate for the conversion).

4.4.2. Minimum required target area

The minimum required target area measures the minimum target area visible on the screen for the AR framework to still track the target (after initial recognition).

To measure the minimum required target area we analysed video from movement #1 on both the left and right motions. For each motion we froze the video frame just after the overlay image disappeared (or just after a clear "jump" in the image, signalling that the framework lost tracking) and marked the original target image with the visible target area (see Figure 7). This was done "by hand" and represents and approximation. We then calculated the visible area as a percentage of the total target area.

(a) *Frame after loss of tracking* **(b)** *Visible target area highlighted (checkerboard not visible)*

Figure 7: *Minimum required target area analysis.*

4.4.3. Visual alignment and stability

Visual alignment measures how close to perfection the alignment between the target and the overlay images is.

We analysed videos from movements #2 and #3 and performed a subjective evaluation of the visual alignment using the overlay images superimposed on the target images as reference. Both authors performed the evaluation independently, using a 3-item scale [-1, 0, 1] corresponding to "bad alignment" – the overlay is grossly aligned with the target, "ok alignment" – there are visible but minor misalignments, "good alignment" – it is hard to tell if there is any misalignment.

Visual stability measures how much the overlay moves in relation to the target, i.e., as the camera moves is the overlay stable relative to the target (even if misaligned), or does it have a "jerky" motion.

We analysed videos from movements #2 and #3 and performed a subjective evaluation of the visual stability. Both authors performed the evaluation independently, using a 3-item scale [-1, 0, 1] corresponding to "very unstable" – the overlay is visible unstable relative to the target, "stable" – there are visible but minor jerkiness motion, "very stable" – it is hard to tell if there is any motion of the overlay relative to the target.

5. Results and Discussion

Videos recorded [BC18b] during the experiment session were downloaded and analysed.

5.1. Overall recognition

Not all mosaics were recognized by all AR frameworks. Additionally, some videos were not correctly recorded even though the corresponding framework did recognize the mosaic. Table 3 lists all

the mosaics in the experiment and whether they were successfully recognized. In addition, the table lists the missing videos due to recording error.

Table 3: *Overview of detection results for each AR framework. The table shows whether the framework succeeded in recognizing the mosaic during the experiment. It also shows the missing video numbers in parenthesis.*

Mosaic	CraftAR	PixLive	Wikitude
Mosaico 1_c	Yes	Yes	No
Mosaico 3_e	No	No	No
Mosaico 6_d	Yes	Yes	No
Mosaico 9_f	No	No	No
Mosaico 10_b	Yes	No	No
Mosaico 13_f	Yes	No	No
Mosaico 14_g	Yes (2)	Yes	No
Mosaico 15_f	Yes	No	No
Mosaico 19_a	No	No	No
Mosaico 20_b	Yes	No	No
Mosaico 21_a	Yes (3)	No	No
Mosaico 22_e	Yes	No	No
Mosaico 23_b	No	No	No
Mosaico 24_d	No	No	No
Mosaico 25_c	Yes	Yes	No
Mosaico 26_d	Yes	Yes	Yes
Mosaico 27_a	Yes	Yes	No
Mosaico 28_d	Yes	Yes	Yes
Mosaico 29_b	Yes	No	No
Mosaico 31_a	Yes (1,2,3)	Yes (1,2,3)	No
"Yes" Count	15	8	2

Overall, CraftAR was able to recognize 15 mosaics out of 20 (75%), against 8 (40%) recognized by PixLive, and only 2 (10%) by Wikitude. These results are doubly surprising to us.

On the one hand, we were not expecting that one of the frameworks would achieve such a high rate of recognition. Even though most of the target images were highly rated by CraftAR, our expectation was that this would be true for the printed images of the mosaics, not for the mosaics themselves. Given the adverse context of use for AR (outdoors, real mosaics, lighting conditions, unsophisticated method for target image acquisition), we were expecting a lower recognition rate.

On the other hand, given previous informal experience with Wikitude recognizing traditional Portuguese sidewalk pavement images, we were expecting it to perform better with Roman mosaics. Additionally, most target images were also highly rated by the Wikitude target manager interface. The Wikitude results should be explained with further research in the future, but given the low number of detected mosaics, Wikitude was excluded from further analysis.

Eight mosaics were successfully recognized by both CraftAR and PixLive so the results of the following sections use only the data collected from these 8 mosaics (actually 7, because all the videos for the last mosaic were incorrectly recorded).

5.2. Recognition delay

Recognition delays for CraftAR and PixLive are shown in Table 4 with boxplots in Figure 8.

It should be noted that these values are not true target recognition delays and cannot be compared, for example, with the study by Marneanu et al. [MER14] in which recognition delay was measured in a more precise way by having the measurement start at a point where the target was already visible to the application. However, as a comparison tool, these values are still useful, and allow us to determine that CraftAR is able to detect targets slightly faster than PixLive (about half a second on average). CraftAR seems to be able to detect the target even with camera movement since most of the delays are negative.

Table 4: *Recognition delay (seconds).*

Mosaic	CraftAR	PixLive
Mosaico 1_c	-0.29	1.44
Mosaico 6_d	-0.21	0.17
Mosaico 14_g	-0.09	0.47
Mosaico 25_c	-0.07	0.18
Mosaico 26_d	0.56	0.15
Mosaico 27_a	-0.14	0.56
Mosaico 28_d	-0.08	0.32
Mean	-0.05	0.47
SD	0.28	0.46

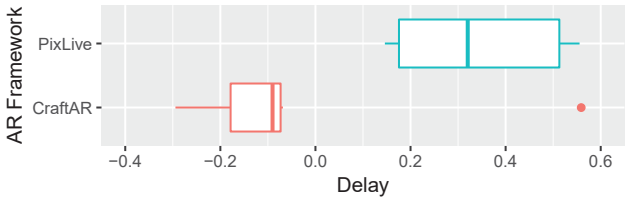


Figure 8: *Detection delay.*

5.3. Minimum required target area

The minimum required target area to keep tracking was an average of 40% (16% SD) for CraftAR and 20% (12% SD) for PixLive (see Figure 9).

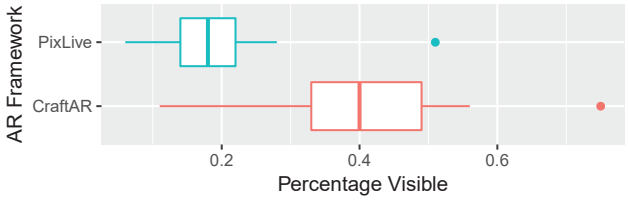


Figure 9: *Minimum required target area for tracking.*

Again, these results cannot be directly compared to the study by Marneanu et al. [MER14], in which they measured the recognition

time for different levels of visibility of a target. We are measuring levels of visibility of a tracked target (after initial recognition with a fully visible target). In their study, CraftAR was able to detect targets even as low as 20% visibility. In ours, CraftAR was able to track targets to about 40% visibility, but under a much less controlled environment.

Our results indicate that PixLive seems to require less visible target area to track the target. This may be a relevant factor in deciding which framework to use, since it may impact (even if slightly) the final user experience. Given that users are not aware of the boundaries of the target being tracked, providing extra freedom in moving the camera is a positive aspect.

5.4. Visual alignment and stability

The authors rated all matching videos between CraftAR and PixLive that resulted from the video recording phase. Since some videos were incorrectly recorded, this resulted in 26 videos for subjective analysis. All videos were rated for visual alignment and stability.

Global percent agreement between the two authors when rating the videos for visual alignment and agreement was 71% (69% for visual alignment rating, and 73% for visual stability). We calculated Weighted Cohen's Kappa to determine if there was agreement between two evaluators. There was moderate overall agreement (using McHugh [McH12] interpretation table), K = 0.76 (95% CI, 0.65 to 0.88). (For alignment ratings only, K = 0.77 (95% CI, 0.61 to 0.92); for stability ratings only, K = 0.75 (95% CI, 0.59 to 0.92)).

Table 5 shows the average ratings for visual alignment and stability for the two frameworks. Is is apparent from Figure 10 that each framework performs better on one parameter and worse on the other.

A Kruskal-Wallis H test showed that there was a statistically significant difference in ratings between CraftAR and PixLive for both visual alignment ($\chi^2(1) = 16.829, p = 4.09e-05$) and visual stability ($\chi^2(1) = 21.803, p = 3.021e-06$).

CraftAR performs better on the visual stability parameter, with an average rating of 0.35 (42% of the CraftAR videos were rated with 1 – very stable – and 50% were rated with 0 – stable). PixLive performs better on the visual alignment parameter, with an average rating of 0.38 (50% of the PixLive videos were rated with 1 – good alignment – and 38% were rated with 0 – ok alignment).

Table 5: *Subjective rating averages.*

	CraftAR	PixLive
Visual alignment	-0.54 (SD 0.65)	0.38 (SD 0.7)
Visual stability	0.35 (SD 0.63)	-0.62 (SD 0.5)

5.5. Summary

Table 6 presents a summary of the comparison between CraftAR and PixLive. CraftAR performs better on the number of recognized mosaics, the target recognition delay and on the visual stability.

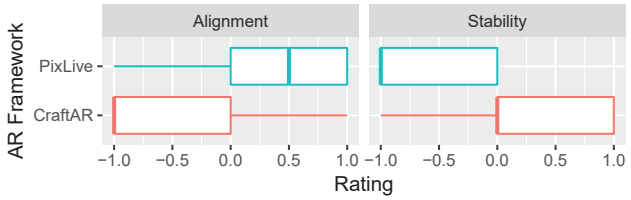


Figure 10: *Subjective ratings – boxplots.*

PixLive performs better on the minimum required target area for tracking and on the visual alignment.

It is interesting to note that these two AR frameworks have been developed with clear different compromises on the implementation of the recognition parameters.

Although CraftAR recognizes almost twice the number of mosaics in our experiment, we believe this should not be the single parameter in deciding between the two. Our target image acquisition process can be much improved and this could have an impact on the recognition rate for PixLive.

Ignoring the overall recognition rate, each framework excels in two parameters: CraftAR achieves better recognition delay and visual stability, while PixLive can maintain tracking with lower target areas in view and achieves better visual alignment.

Table 6: *Summary of comparison between CraftAR and PixLive.*

Criteria	CraftAR	PixLive
Overall recognition rate (n^o of targets)	**15**	8
Detection delay (seconds)	**-0.05**	0.47
Minimum target area (% of target)	40%	**20%**
Visual alignment (avg. rating [-1, 1])	-0.54	**0.38**
Visual stability (avg. rating [-1, 1])	**0.35**	-0.62

6. Conclusion

In the context of the development of a hybrid mobile application for AR experiences with roman mosaics in the Conímbriga site, we have analysed existing AR frameworks. This analysis resulted in three potential frameworks – CraftAR, PixLive, and Wikitude – that were subsequently evaluated in a real setting scenario. We compared the three frameworks for their overall target recognition rate, recognition delay, minimum required target area for tracking, visual alignment and visual stability.

Wikitude failed recognizing most of the configured target images of mosaics. However, given our earlier experiences with Wikitude, we take this result carefully and recognize the need for further research to explain these results. CraftAR was able to recognize 75% of the targets, while PixeLive recognized 40% of the targets.

Regarding the recognition delay, minimum required target area, and visual alignment and stability there was no clear winner. Both CraftAR and PixLive excel at two of these parameters.

These results allow us to understand the strong and weak points

of these AR frameworks when applied to the recognition of Roman mosaics. Clearly, a compromise will have to be made as there is no single framework that achieves the best results in all parameters. Knowing at which parameters each framework performs best will allow us to make informed decisions regarding the choice of framework for application development.

It is clear from this study, that different AR frameworks may have very different behaviour on the same image type. CH practitioners intending to develop AR applications should compare different AR frameworks on their final subject images before committing resources on a specific AR framework.

Although we did not explicitly study this aspect, it is also apparent from our experiment that the quality ratings of images provided by the AR frameworks cannot be completely trusted (at least in some scenarios), as was the case with the Wikitude framework which provided good ratings for the configure targets but failed recognizing most of them in the real environment. Real environment testing should be performed to assess the real quality of the captured targets.

References

[AG15] AMIN D., GOVILKAR S.: Comparative Study of Augmented Reality Sdk's. *International Journal on Computational Science & Applications 5*, 1 (feb 2015), 11–26. URL: http://wireilla.com/papers/ijcsa/V5N1/5115ijcsa02.pdf, doi:10.5121/ijcsa.2015.5102. 3

[Aug18] AUGMENT: Official Augment site. http://www.augment.com/, 2018. URL: http://www.augment.com/. 4

[BC18a] BELO A., CARDOSO J. C. S.: Mobile AR Frameworks. Dataset, 2018. URL: https://figshare.com/articles/Mobile{_}AR{_}Frameworks/5926498, doi:10.6084/m9.figshare.5926498.v1. 3

[BC18b] BELO A., CARDOSO J. C. S.: Mosaic AR application video recordings. Dataset, 2018. doi:10.6084/m9.figshare.5987524. 7

[BC18c] BELO A., CARDOSO J. C. S.: Mosaic Overlays. Dataset, 2018. doi:10.6084/m9.figshare.5928784. 6

[BC18d] BELO A., CARDOSO J. C. S.: Mosaic Targets. Dataset, 2018. doi:10.6084/m9.figshare.5928775. 5

[Cat18] CATCHOOM: Official CraftAR site. https://catchoom.com/product/craftar/augmented-reality-and-image-recognition/, 2018. URL: https://catchoom.com/product/craftar/augmented-reality-and-image-recognition/. 3, 4

[con17a] CONTRIBUTORS W.: Conímbriga — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Con%C3%ADmbriga, 2017. URL: https://en.wikipedia.org/w/index.php?title=Con{í}mbriga. 1

[con17b] CONTRIBUTORS W.: Roman mosaic — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Roman_mosaic, 2017. URL: https://en.wikipedia.org/w/index.php?title=Roman{_}mosaic. 1

[eT18] EZAR TECHNOLOGIES: Official ezAR site. https://www.ezartech.com/, 2018. URL: https://www.ezartech.com/. 4

[FSL05] FRITZ F., SUSPERREGUI A., LINAZA M.: Enhancing cultural tourism experiences with augmented reality technologies. *The 6th International Symposium on Virtual Reality Archaeology and Cultural Heritage VAST* (2005), 20–21. URL: http://public-repository.epoch-net.org/publications/VAST2005/shortpapers/short2005.pdf. 1

[HGT17] HERPICH F., GUARESE R. L. M., TAROUCO L. M. R.: A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications. *Creative Education 08*, 09 (2017), 1433–1451. URL: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ce.2017.89101, doi:10.4236/ce.2017.89101. 2, 3

[Lay18a] LAYAR: Official Layar site. https://www.layar.com/, 2018. URL: https://www.layar.com/. 4

[Lay18b] LAYAR: PhoneGap Layar Plugin. https://www.layar.com/documentation/layar-sdk/phonegap-layar-plugin/, 2018. URL: https://www.layar.com/documentation/layar-sdk/phonegap-layar-plugin/. 4

[Mar17] MARTO A. G. R.: *Realidade Aumentada Móvel num Contexto de Herança Cultural*. PhD thesis, University of Porto, 2017. 3

[McH12] MCHUGH M. L.: Interrater reliability: the kappa statistic. *Biochemia medica 22*, 3 (2012), 276–82. URL: http://www.ncbi.nlm.nih.gov/pubmed/23092060http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3900052. 9

[MER14] MARNEANU I., EBNER M., ROESSLER T.: Evaluation of Augmented Reality Frameworks for Android Development. *International Journal of Interactive Mobile Technologies (iJIM) 8*, 4 (2014), 37. URL: http://online-journals.org/index.php/i-jim/article/view/3974, doi:10.3991/ijim.v8i4.3974. 3, 8

[NSP09] NOH Z., SUNAR M. S., PAN Z.: A Review on Augmented Reality for Virtual Heritage System. In *Proceedings of the 4th International Conference on E-Learning and Games: Learning by Playing. Game-based Education System Design and Development* (2009), Springer-Verlag, pp. 50–61. URL: http://link.springer.com/10.1007/978-3-642-03364-3{_}7, doi:10.1007/978-3-642-03364-3_7. 1

[RCJ16] RAUTENBACH V., COETZEE S., JOOSTE D.: Results of an evaluation of augmented reality mobile development frameworks for addresses in augmented reality. *Spatial Information Research 24*, 3 (2016), 211–223. URL: http://link.springer.com/10.1007/s41324-016-0022-1, doi:10.1007/s41324-016-0022-1. 3

[SGM*15] SCOPIGNO R., GABELLONE F., MALOMO L., BANTERLE F., PINGI P., AMATO G., CARDILLO F. A.: LecceAR: An Augmented Reality App. In *Digital Presentation and Preservation of Cultural and Scientific Heritage* (2015), pp. 99–108. 1

[Soc18] SOCIALCOMPARE: Augmented Reality SDK Comparison. http://socialcompare.com/en/comparison/augmented-reality-sdks, 2018. URL: http://socialcompare.com/en/comparison/augmented-reality-sdks. 3

[Vid18] VIDINOTI: Official PixLive site. https://www.vidinoti.com/, 2018. URL: https://www.vidinoti.com/. 4

[VIK*02] VLAHAKIS V., IOANNIDIS N., KARIGIANNIS J., TSOTROS M., GOUNARIS M., STRICKER D., GLEUE T., DAEHNE P., ALMEIDA L.: Archeoguide: An augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications 22*, 5 (2002), 52–60. URL: http://ieeexplore.ieee.org/document/1028726/, doi:10.1109/MCG.2002.1028726. 1

[Wik18] WIKITUDE GMBH: Official Wikitude site. https://www.wikitude.com/, 2018. URL: https://www.wikitude.com/. 4