ARKit and ARCore in serve to augmented reality

Zainab Oufqir
LIIAN, Department of Mathematics and informatic
USMBA Faculty of Sciences Fes, Morocco
zayna.oufkir@gmail.com

Abdellatif EL ABDERRAHMANI
LIIAN, Department of Mathematics and informatic
USMBA Faculty of Sciences Fes, Morocco
elabderrahmani@yahoo.fr

Khalid Satori
LIIAN, Department of Mathematics and informatic
USMBA Faculty of Sciences Fes, Morocco
khalidsatori@gmail.com

***Abstract*— Many libraries are available in the development world to create augmented reality applications, their functionality differs depending on the technology used to detect and track an object, points or features in a scene. In this article, we will discover ARKit, ARCore two open source libraries that display a virtual object in the real world. Their goal is to merge digital content and information with the real world. They can interact with the components of the device (camera and screen) to detect and track characteristics of the scene in order to insert virtual content. This study implements and concretizes the different functionalities available in augmented reality to enrich the real world with additional information.**

***Keywords*— arkit, arcore, augmented reality, real world, virtual 3D object**

I. Introduction

Augmented reality refers to methods and technologies for inserting virtual objects into a sequence of real images acquired through the camera of a device in a realistic way [1]. Unlike virtual reality, where the user is fully immersed in a digital world, augmented reality aims to add some virtual content to the real world and offers the user real-time interaction possibilities, and is generally based on a 3D environment. The objective of virtual objects is to provide relevant and contextualized information giving a new meaning to the user of the Augmented Reality system. This combination is possible by using electronic devices containing a camera sensor (glasses, smartphone, tablet, etc.). Smartphones and tablets are currently the most widely used, virtual images are superimposed on the reality filmed by the camera of the tablet or smartphone [2].

The main objective for an augmented reality system is to know the pose of the camera recording the scene in relation to a known object in the scene. The objective of the pose estimation is to find the translation and rotation between a known object in the real world and the camera [3]. It is assumed that the intrinsic parameters (focal length and main point) of the camera are known [4], i.e. the camera is calibrated, this problem is solved by giving a list of correspondences between known 3D points of the object and their 2D projections on the camera screen [5]. There are two techniques in Augmented Reality to calculate the pose, marker-based and markerless methods [6]. A Marker-based application requires the prior registration of an image for it to be recognized. This method is the most widely used because the markers are easy to recognize and track, and does not require powerful devices [7] [8]. A Markerless application recognizes objects without having been registered beforehand. The implementation of this method is quite complicated. The algorithm that identifies the objects must recognize different things, such as patterns, colors and other characteristics that exist in the scene [9] [10]. Today's applications tend most towards markerless methods due to rapid advances in technology. For several years now, augmented reality has been part of our daily lives, such as Snapchat or Messenger, the IKEA catalogue or the Pokemon GO! game, and there is also several scientific publications using this technology [11] [12] [13] [14]. To integrate Augmented Reality into a mobile application, a specific SDK (Software Development Kit) must be used. These SDKs exist in large quantities on the market, they all have their own specifications. The choice is made according to the needs and according to certain criteria. The main SDKs for augmented reality present on the market today offer interesting possibilities that allow to project oneself in the conception of an application for the industry. The arrival of Augmented Reality Development Kits, Apple's ARKit for iOS, Google's ARCore for Android allow developers to easily create Augmented Reality applications, the interest of these platforms is mainly to considerably reduce the difficulty of access to Augmented Reality for developers.

In the following sections, we will first present the two libraries with their main functionalities, the different versions, the compatible devices and tests on each functionality, then an interpretation that presents a solution

I. ARKIT

ARKit is Apple's Augmented Reality Application Development Kit released in June 2017. This kit is dedicated to iOS and allows to create augmented reality applications for iPhone and iPad under iOS 11 or higher. The Apple ARKit SDK is available for download for all iOS developers who have an Apple Developer Account.

A. Main features

Tracking: Apple ARKit allows you to track the smartphone's position in the real world in real time. To do this, the SDK combines the VIO (Visual Inertial Odometer) with data from the camera and the motion sensor of the device.

Understanding the environment: With ARKit, iPhone and iPad can identify real-world vertical and horizontal flat surfaces in real time for realistic integration of virtual elements.

Static 2D image recognition, this feature allows to store images in the application and to detect them in a real environment, to display and superimpose virtual information on these images.

978-1-7281-8041-0/20/$31.00 ©2020 IEEE

ARKit is also able to use the brightness sensor to determine the amount of light in the surrounding room to apply the degree of brightness to the virtual object. ARKit can also create reflections of real objects on virtual metal.

B. Versions

ARKit 2 provides new contextual elements for augmented reality experiences. Thus, this new version has brought the following new functionalities [1] :

- Detect static 3D objects using ARKit Scanner to scan the object defined in the yellow box (fig1) and compress the extracted information into a file with an arobject extension, which then allows the 3D object to be recognized in the real world and augmented with virtual information [16].

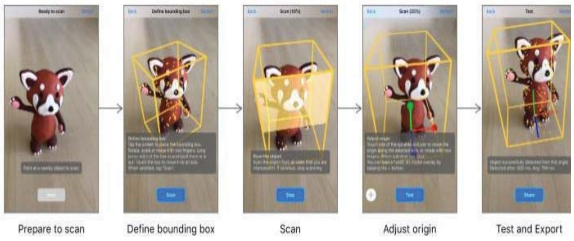

Fig. 1. Scanning 3D object

- **Tracking of moving images** in the scene, this feature allows the virtual object inserted on the detected image to follow the moving image in the scene [17].
- **Face Tracking** follow the movements and facial expressions using the front device of the device, the use of this technology requires an IOS device that integrates the TrueDepth camera as iPhone X and iPhone XS [18].
- **Save a Map of the environment**, during a first augmented reality experience in an unknown environment, the user can save a card containing all the information of this environment and reload this card in the same environment which allows to optimize the detection and tracking process by using the already recorded information of this scene [19].
- **MiltiUser** in augmented reality applications or games: Augmented reality experiences can now be shared and fixed in real locations between two or more users [20]. At the keynote, Apple presented the Galaga application that integrates this feature, a multiplayer augmented reality game developed by the Directive Games studio that allows players to engage in war machine combat.

Apple ARKit 3 is the current version announced at WWDC 2019, it introduces many major innovations for augmented reality [7] :

- **Tracking of the body and occlusion**, this feature includes the position of people in the space visible to the camera and to place virtual objects in augmented reality in front of or behind the people in a realistic way.
- **Motion Capture**, it allows the tracking of a virtual copy in the form of a skeleton of the person filmed by the device, which makes it possible for example to reproduce body movements on an avatar for video games or other types of applications.
- **Multiple Face Tracking**, it can detect and track three or more faces at the same time with cameras that support the TrueDepth camera.
- **Simultaneous Front and Back Camera**, ARKit 3 will enable simultaneous front and back camera capture, motion capture, faster reference image loading, automatic image size detection and better visual consistency.
- **Additional features**: The ability to detect 100 images at the same time and be able to estimate their size automatically, improved 3D object detection with fast detection in complex environments, and the use of the machine learning to improve the detection quality of flat surfaces.

The following figure summarizes the devices compatible with the different versions of ARKit:

Fig. 2. Supported devices for ARKit 2 & 3

For virtual objects inserted in the scene, ARKit supports several 3D formats with the following extensions: .abc .usd(.usda, .usdc) .usdz .ply .obj .stl [22], other formats can be supported after.

C. Tests

To test ARKit, we chose to work with the iPhone 8, using the XCode editor for development [23] and the development language used is Swift [24]. Apple does not offer the possibility to test applications on simulators, it is necessary to have a compatible device [25].

a) Feature points

ARKit builds a cloud of points representing its approximate understanding of the 3D world around the user. Figure 3 shows these yellow points of key points extracted by ARKit, they represent the characteristics of the scene detected in the images acquired in real time by the camera. Their positions in the 3D world coordinate space are extrapolated as part of the image analysis that ARKit performs in order to accurately track the position, orientation and movement of the camera, the more points there are, the better the position tracking is.

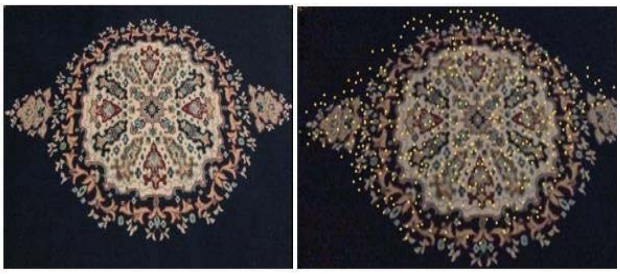

Fig. 3. Feature points (yellow dots)

a) Plane detection

ARKit detects horizontal and vertical planes through points of interest extracted and tracked in the scene. These points can be corners, lines, changes in color or shape, edges of objects, etc. To demonstrate the detection of a plane, the application visualizes the estimated shape of each detected plane and draws a blue delimitation rectangle for it as shown in the figure 4:

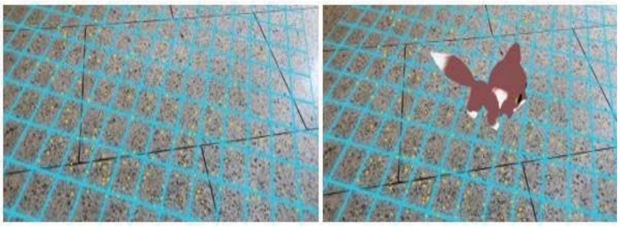

Fig. 4. Plan delimited with blue boxs and augmented with 3D object

ARKit constantly analyzes the scene, when it detects a larger or smaller plane, it updates the values of the rectangle according to the size of the detected plane. Each plane provides information on the estimated position and shape of the surface, which allows a virtual object to be correctly positioned in the scene as shown in the figure 4.

b) Image Tracking

ARKit 2 has introduced an image tracking function, it can detect static or dynamic moving images in a scene. The image is stored beforehand in the application, and by analyzing the scene ARKit detects the image and inserts a virtual 3D object on it as shown in the figure 5.

Fig. 5. : Image detected and augmented with a virtual object

c) Object Detection

ARKit 2 also offers the possibility to detect real-world 3D objects. The 3D object must be scanned to extract the characteristic points and store them in a file with an arobject extension using the ARKit Scanner application provided by Apple [16] as shown in the figure 6:

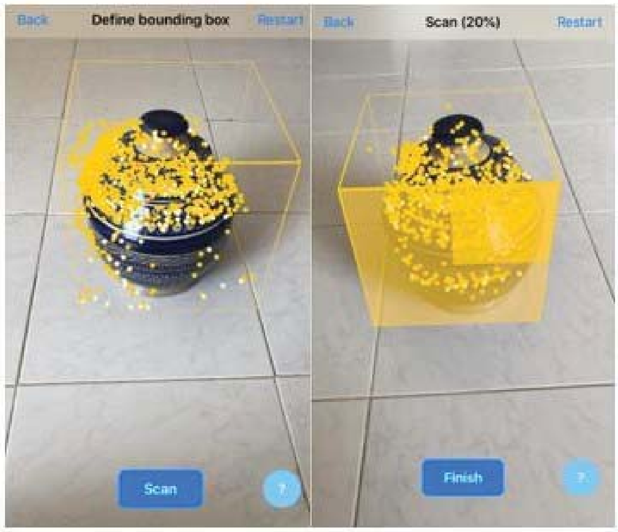

Fig. 6. Scanning a 3D object

The object is surrounded by a yellow rectangle, the application detects the characteristic points (the more points extracted the better the object recognition) and scans all areas of the rectangle. ARKit uses the information extracted from the arobject file provided by the ARKit Scanner application to detect the object in different scenes and augmented it with additional information as shown in the figure 7:

Fig. 7. 3D object detected in different scene and augmented with text

b) Face detection

Face detection is limited on devices with Face ID (IPhone X, IPhone XS ...), ARKit follows facial expressions, facial geometry, appearance, intensity, temperature and direction of light on the face. [18]. To work with facial expressions in ARKit, there are about 50 position coefficients for parts of the face (eyebrows, nose, cheeks, lips, tongue) [26], it can transfer facial expressions and head movements to 2D and 3D objects, creating animated characters based on the movements and facial expressions of real people [27].

II. ARCORE

In 2014, Google launched its first Tango SDK dedicated to Augmented Reality, it required a smartphone equipped with a depth sensing camera to work. Tango's particularity was to allow its users to map environments in 2D and 3D. Lenovo phab 2 pro was the first smartphone to integrate Tango technology, enabling its users to benefit from functionalities

related to augmented reality. This project did not last very long because it was limited to devices with essential sensors that no smartphone had by default. Google is relaunching the adventure of augmented reality by launching the ARCore project in March 2018. Unlike Tango, ARCore is usable on several smartphones with Android Nougat (7.0) and IOS devices that support ARKit. ARCore is based on the Tango project with fewer features that do not depend on specialized cameras, allowing a larger number of devices to be compatible with its technology. We find here the same technologies as on ARKit, namely motion tracking, space understanding and brightness estimation.

A. *Main Features*

Tracking: ARCore uses the phone's camera to observe the characteristic points of the scene and the data from the Inertial Measurement Unit (IMU) sensor, it determines the position and orientation of the device as it moves. This allows the virtual objects to be properly positioned.

Understanding the environment: ARCore uses the different sensors in the smartphone to detect horizontal surfaces using the same feature points it uses for motion tracking.

Brightness estimation: The light estimation that allows the smartphone to estimate the ambient light in order to perfectly integrate the 3D elements into the real world.

B. *Versions*

The different versions of ARCore that have brought great changes in functionality [28]:

- version 1.0: World tracking and horizontal plane detection
- version 1.2: Allows ARCore applications to detect images and add the ability to detect vertical surfaces
- version 1.7: Enable Augmented Faces with the front-facing (selfie) camera
- version 1.9 : tracking moving images in the scene with the possibility to follow the images even if they are no longer visible by the camera. ARCore can detect up to 20 images at a time [29]
- version 1.10: New Environmental HDR mode added to Light Estimation API
- version 1.11: ARCore now targets 60fps on supported devices. Use new camera config filters to target 30fps on all ARCore devices
- The other versions have corrected various bugs detected in ARCore or made improvements on previous features.

The 3D file formats supported by ARCore are: .obj .glTF .fbx [30]. These 3D formats are created by software such as Blender [31] or SketchUp [32], they provide the different 3D formats supported by ARKit and ARCore.

C. *Tests*

To test ARCore, we chose to work with the Samsung Galaxy S8 smartphone, using Android Studio for development. [33] and Java as a programming language.

a) Feature points

ARCore scans the environment and detects visually distinct entities in the images called key points and uses these points to calculate its change of location. These points are represented by a green color in the figure 8.

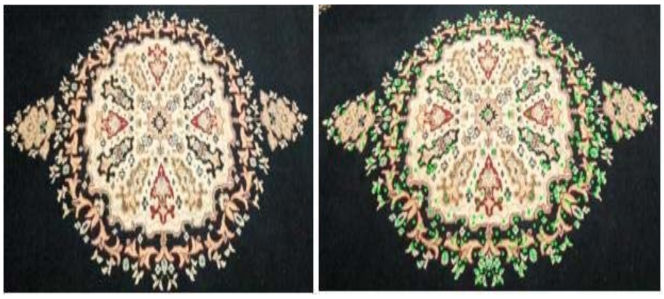

Fig. 8. Feature points (green dots)

b) Plane detection

ARCore searches in the scene for groups of key points that can build horizontal and vertical plans, it can also determine the boundaries of each plan. Figure 9 shows the plane detected in a scene, this plane is delimited by white dots and augmented by a 3D virtual object.

Fig. 9. Plan delimited with white points and augmented with 3D object

c) Image Tracking

Users can create Augmented Reality experiences by pointing their phone camera at specific images that have been previously saved in the application database. Figure 10 shows a scene containing an image, the application scans the scene and detects the image by inserting a 3D virtual object on it.

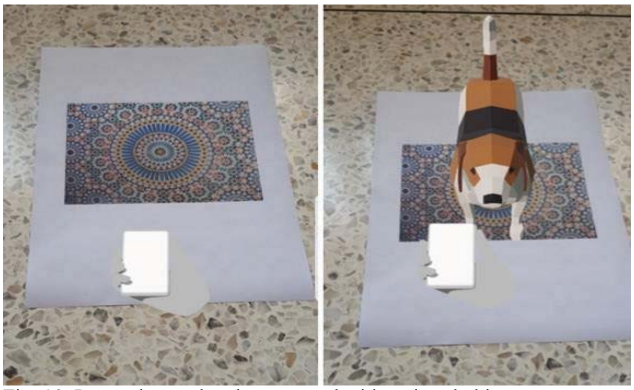

Fig. 10. Image detected and augmented with a virtual object

d) Object detection

One of ARCore's limitations is that it can only recognize 2D markers, unlike ARKit, which can scan and recognize static 3D objects.

e) Face detection

ARCore Augmented Faces allows you to track the geometry of the face, the position of the tip of the nose, the centre of the head, the left and right sides of the forehead, this feature is available on all devices that support ARCore [34]. In order to correctly overlay textures and 3D models on a detected face (Figure 11), ARCore provides detected regions and an increased face mesh. This mesh is a virtual representation of the face and consists of the vertices, facial regions and the center of the user's head [34].

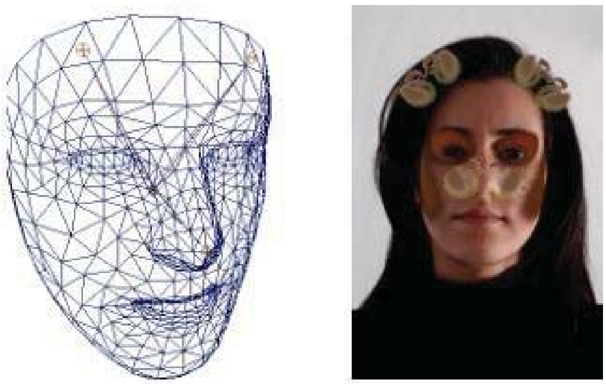

Fig. 11. Face detected and augmented with a virtual object

III. INTERPRETATION

ARKit and ARCore share the following features to insert a virtual object in the real world:

- Motion tracking: track the position of the device in the real world
- Planes detection: detection of horizontal and vertical surfaces
- Light estimation: to detect information about the lighting
- Image tracking: detect and track an image in a scene
- Face detection: detect and track a face in a scene

Both libraries aim to build augmented reality applications to enrich the real world with additional 2D or 3D digital information using a smartphone or tablet and both platforms have their strengths and weaknesses [35]. The choice of the technology to be used to develop an augmented reality application depends on the needs and on different criteria, a simple search on the number of projects done on GitHub and the number of questions asked on Stackoverflow shows the following result:

December 2019	**ARKit**	**ARCore**
Github repository results	3720	1280
Stackoverflow question threads	2246	1195

The results show the preference of ARKit by the developers. Questions are asked twice more in Stackoverflow and the number of projects carried out on GitHub and three times more compared to ARCore. These results lead us to ask a question. why ARKit is much more used than ARCore when there are 85% of Android smartphones on the market and the number of devices compatible with ARCore is greater than that of ARKit which are limited to IOS devices. Looking at the hardware side, Apple's devices are fully compatible with ARKit, they have included the same sensors in their device and they have already preconfigured their devices to support augmented reality applications. Unlike the Android device, the hardwares and sensors are different from one device to another and the user must install the ARCore application from google play [36] to support augmented reality applications and some new ARCore-compatible devices do not support the installation of this application when released. For the number of images processed per second, ARKit processes 60 images per second [37], ARCore is limited to 30 frames per second but can process 60 frames per second for devices that support it [38] which explains the limitation of ARCore on the hardware side.

Development environment:

Augmented reality is based on image processing and 3D rendering methods. To develop augmented reality applications, it is necessary to work with software that allows 3D rendering. The following table shows the compatibility of Augmented Reality SDKs with different development environments.

	ARKit	**ARCore**
Android Studio	No	Yes
XCode	Yes	No
Unity	Yes	Yes
Unreal engine	Yes	Yes

Unity has introduced ARFoundation [39] which brings together the functionalities of ARKit and ARCore in a single module, which allows the developer to target both IOS and Android platforms with the same source code. This allows an application to be developed once and deployed on both devices without any modification. The following table lists the features currently supported in ARFoundation [40]:

Supported Feature	AR Foundation	Google ARCore SDK for Unity	Unity ARKit Plugi
Plane Detection (Vertical)	✔	✔	✔
Plane Detection (Horizontal)	✔	✔	✔
Feature Point Detection	✔	✔ + Oriented Feature Points	✔
Light Estimation	✔	✔ + Color Correction	✔ + Color Temperatu
Hit Testing (Feature point and Plane raycasting)	✔	✔	✔
Image Tracking	in development	✔ (Static Only)	✔
3D Object Tracking	in development		✔
Environment Probes	in development		✔
World Maps	✔		✔
Face Tracking (Pose, Mesh, Blendshapes)	✔		✔ iPhone X + Variants
Cloud Anchors	in development	✔	

Fig. 12. ARFoundation features

As shown in the table, AR Foundation does not yet implement all the features of ARKit and ARCore, so if an application depends on a specific feature that is not yet in AR Foundation, these specific SDKs must be used separately.

I. Conclusion

In this article, we presented the two libraries ARKit and ARCore and their capability in the development of augmented reality applications. ARCore is the framework for Android, while ARKit is the one for Apple's iOS. The interest of these platforms is mainly to reduce considerably the difficulty of access to augmented reality for developers. These frameworks allow developers to use advanced augmented reality tools to enable an enhanced, more immersive and realistic augmented reality. The choice of the technology to be used depends essentially on the needs and the functionalities to be implemented. With these technology leaders, Augmented Reality has become accessible to the general public through their smartphones and allows them to integrate real-time 3D virtual elements into their environments.

References

[1] R. T. Azuma, « A Survey of Augmented Reality », p. 48, 1997.

[2] D. Chatzopoulos, C. Bermejo, Z. Huang, et P. Hui, « Mobile Augmented Reality Survey: From Where We Are to Where We Go », IEEE Access, vol. 5, p. 6917□6950, 2017, doi: 10.1109/ACCESS.2017.2698164.

[3] Z. Oufqir, A. E. Abderrahmani, et K. Satori, « Important Method for Detecting and Tracking Based on Color », vol. 8, no 5, p. 5, 2019.

[4] E. Marchand, H. Uchiyama, et F. Spindler, « Pose Estimation for Augmented Reality: A Hands-On Survey », IEEE Transactions on Visualization and Computer Graphics, vol. 22, no 12, p. 2633□2651, déc. 2016, doi: 10.1109/TVCG.2015.2513408.

[5] M. Billinghurst, A. Clark, et G. Lee, « A Survey of Augmented Reality », FNT in Human–Computer Interaction, vol. 8, no 2□3, p. 73□272, 2015, doi: 10.1561/1100000049.

[6] Z. Oufqir, A. E. Abderrahmani, et K. Satori, « From marker to markerless in augmented reality », p. 14.

[7] H. Kato et M. Billinghurst, « Marker tracking and HMD calibration for a video-based augmented reality conferencing system », in Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99), 1999, p. 85□94, doi: 10.1109/IWAR.1999.803809.

[8] [8] M. Hirzer, « Marker Detection for Augmented Reality Applications », p. 28.

[9] J. P. Lima, F. Simões, L. Figueiredo, et J. Kelner, « Model Based Markerless 3D Tracking applied to Augmented Reality », vol. 1, p. 14, 2010.

[10] P. Honkamaa, S. Siltanen, J. Jäppinen, C. Woodward, et O. Korkalo, « Interactive outdoor mobile augmentation using markerless tracking and GPS », p. 5.

[11] R. Bergquist et N. Stenbeck, « Using Augmented Reality to Measure Vertical Surfaces », p. 12.

[12] C. T. Tan et D. Soh, « AUGMENTED REALITY GAMES: A REVIEW », p. 9.

[13] R. Cervenak et P. Masek, « ARKit as indoor positioning system », in 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland, 2019, p. 1□5, doi: 10.1109/ICUMT48472.2019.8970761.

[14] X. Zhang, X. Yao, Y. Zhu, et F. Hu, « An ARCore Based User Centric Assistive Navigation System for Visually Impaired People », Applied Sciences, vol. 9, no 5, p. 989, mars 2019, doi: 10.3390/app9050989.

[15] « What's New in ARKit 2 - WWDC 2018 - Videos », Apple Developer.https://developer.apple.com/videos/play/wwdc2018/602/.

[16] « Scanning and Detecting 3D Objects | Apple Developer Documentationhttps://developer.apple.com/documentation/arkit/scanning_and_detect ing_3d_objects.

[17] « Tracking and Altering Images | Apple Developer Documentation».https://developer.apple.com/documentation/arkit/tracking_and_alterin g_images.

[18] « Tracking and Visualizing Faces | Apple Developer Documentation.https://developer.apple.com/documentation/arkit/tracking_and_visuali zing_faces.

[19] « Saving and Loading World Data | Apple Developer Documentation».https://developer.apple.com/documentation/arkit/saving_and_loading_world_data.

[20] « Creating a Multiuser AR Experience | Apple Developer Documentation».https://developer.apple.com/documentation/arkit/creating_a_multiuser_ar_experience.

[21] « ARKit 3 - Augmented Reality », Apple Developer. https://developer.apple.com/augmented-reality/arkit/.

[22] « canImportFileExtension(_:) - MDLAsset | Apple Developer Documentation». https://developer.apple.com/documentation/modelio/mdlasset/139181 3-canimportfileextension.

[23] « Xcode - Apple Developer »https://developer.apple.com/xcode/.

[24] « Swift - Apple (FR) ». https://www.apple.com/fr/swift/.

[25] « Device Compatibility »https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMa trix/DeviceCompatibilityMatrix.html.

[26] « ARFaceAnchor.BlendShapeLocation - ARFaceAnchor | Apple DeveloperDocumentation». https://developer.apple.com/documentation/arkit/arfaceanchor/blends hapelocation.

[27] « blendShapes - ARFaceAnchor | Apple Developer Documentation.https://developer.apple.com/documentation/arkit/arfaceanchor/292825 1-blendshapes.

[28] «google-ar/arcore-android-sdk»,GitHub. https://github.com/google-ar/arcore-android-sdk.

[29] « Recognize and Augment Images | ARCore », Google Developers. https://developers.google.com/ar/develop/c/augmented-images?hl=fr.

[30] « Import and preview 3D assets | ARCore », Google Developers. https://developers.google.com/ar/develop/java/sceneform/import-assets?hl=fr.

[31] B. Foundation, « blender.org - Home of the Blender project - Free and Open 3D Creation Software », blender.org. https://www.blender.org/.

[32] « 3D Design Software | 3D Modeling on the Web | SketchUp ».

https://www.sketchup.com/.

[33] « Download Android Studio and SDK tools », Android Developers. https://developer.android.com/studio?hl=fr.

[34] « Augmented Faces | ARCore », Google Developers.: https://developers.google.com/ar/develop/java/augmented-faces?hl=fr.

[35] P. Nowacki et M. Woda, « Capabilities of ARCore and ARKit Platforms for AR/VR Applications », in Engineering in Dependability of Computer Systems and Networks, vol. 987, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, et J. Kacprzyk, Éd. Cham: Springer International Publishing, 2020, p. 358□370.

[36] « Google Play Services for AR - Apps on Google Play ». https://play.google.com/store/apps/details?id=com.google.ar.core&hl=en_US.

[37] « Augmented Reality Application », Apple (France). https://www.apple.com/fr/ios/augmented-reality/.

[38] « Configuring the camera | ARCore », Google Developers.https://developers.google.com/ar/develop/unity/camera-configs?hl=fr.

[39] « Framework AR Foundation d'Unity | Unity ».https://unity.com/fr/unity/features/arfoundation.

[40] « Unity's Handheld AR Ecosystem: AR Foundation, ARCore and ARKit».https://blogs.unity3d.com/es/2018/12/18/unitys-handheld-ar- ecosystem-ar-foundation-arcore-and-arkit/.