Quality Model for Testing Augmented Reality Applications

Jim Scheibmeir
Research and Advisory
Gartner
Fort Myers, United States
jimscheibmeir@gmail.com

Yashwant K. Malaiya
Computer Science
Colorado State University
Fort Collins United States
malaiya@cs.colostate.edu

Abstract—**Augmented Reality applications have the capability of merging virtual objects into physical setting, or alternatively they can wrap physical objects within a virtual scene. Augmented reality applications are similar to virtual reality applications in that aspects of the visualizations are computer generated, but augmented reality apps also must contain a view of the physical world. Augmented reality applications are being utilized in service, manufacturing, product areas, as well as gaming. Mobile devices are becoming common runtime environments for augmented reality applications and the mobile device proliferation is enabling a wave of AR applications. Due to the combined nature of digital and physical objects, as well as the environmental and contextual constraints, a traditional test plan is not sufficient. A new quality model is proposed that takes these issues into account, and an example of how machine learning can assist with aspects of the model is discussed.**

Keywords—*Augmented Reality, Software Quality, Software Testing, Machine Learning, Test Automation, Quality Model*

I. INTRODUCTION

Augmented reality is accomplished by creating an experience in an application's user interface that merges both physical surroundings as well virtual objects, which may be models or scenes [1]. The term *augmented reality* was first used by Caudell and Mizell in 1992 [2]. The development of augmented reality applications has very much been democratized within the last decade, due to the rise of libraries that make building the applications easier, and the increase in compute power of devices such as mobile phones. Mobile phones have emerged into ubiquitous platforms as they fit into our lives, rather than requiring human change [3]. Similarly, augmented reality has allowed us to keep a physical plane as-is, and yet embellish upon it new digital objects. AR is becoming popular because it allows change to what we see and at the same time it allows the physical world to stay the same. According to one Gartner survey conducted in 2018 (Figure 1), "According to a Gartner Research Circle study, 27% and 17% of respondents were either using or evaluating AR and MR technologies "[4].

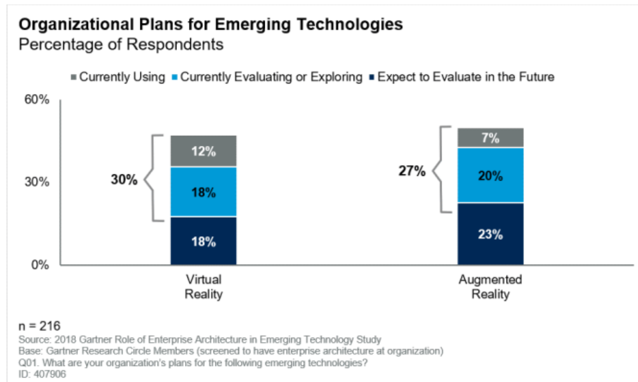

Figure 1. Gartner Emerging Technology Analysis: Augmented and Mixed Reality Opportunity for 3D Design Software and Vertical ISVs

An additional Gartner survey focused on multi-experience application development. The analysts stated, "It is, however, surprising to see VR apps identified as the second most impactful type of multiexperience app (20%), as AR has more potential use cases and device support. But only 14% of the respondents thought that AR apps would be the most impactful, despite AR app development tools being more widely available. " [5] (figure 2).

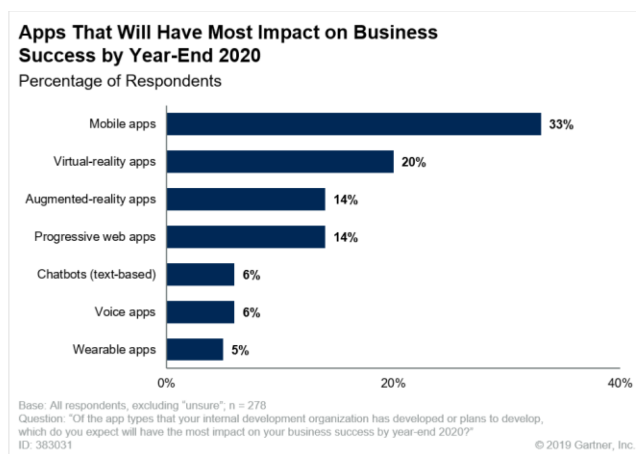

Figure 2. Gartner Survey Analysis: Insights to Kick-Start an Enterprise Multiexperience Development Strategy

There are several development tools to assist developers in the creation of AR applications. The ARToolkit was

978-1-7281-3885-5/19/$31.00 ©2019 IEEE

developed by Hirokazu Kato in 1999 which is now available for Windows, Mac, Linux, and Android platforms. ARCore, an SDK for Android and iOS phone, was developed by Google and released in 2018 [6]. Apple has their own native tooling, ARKit, which released with iOS 11 in 2017 [7]. Facebook released Camera Effects Platform in 2017. In October of 2018, Facebook rebranded the platform to Spark AR [8]. Qualcomm released the QCAR SDK in 2011 [9], it then sold to PTC in 2015 for $65 million [10]. Maxst started as a company in 2010, released an AR mobile game in 2011, 5 years before the release of Pokemon GO, and released their AR SDK in 2012 [11]. Wikitude's Javascript SDK released in 2012 [12].

One driving factor of the popularity of mobile augmented reality applications, are the many SDKs available and the ease of use. The ease of development, however, has also led to applications having poor user experiences [13]. There are several choices of development tools, both open source, and commercial, which span across runtime environments. Table 1 lists several common examples of these types of augmented reality application development tools.

TABLE I. EXAMPLE AUGMENTED REALITY DEVELOPMENT TOOLS

Tool name	Platform
Vuforia Studio [14]	Android, iOS, UWP, Unity
Spark AR [15]	Android, iOS
ARKit [16]	iOS
ARCore [17]	Android, iOS
Wikitude [18]	Android, iOS, Microsoft Tablet, smart glasses
Maxst AR [19]	Android, iOS, Windows, Mac, Unity

Industry usage of augmented reality technology is now becoming wide-spread. AR applications have been utilized in manufacturing, service industry, and product design. AR has aided in the layout and design of factories [20]. Military machinery service and maintenance has also been enabled thru AR technology [21]. AR has also been utilized to assist in training new assembly skills [22]. In the automotive industry, augmented reality was utilized by Mercedes Benz to create a consumer application allowing potential buyers to customize a new car as it would look in their own personal driveway. In the first three months of this application being available to iOS users, there were over thirty-four thousand downloads and an average application rating of 4.3 out of 5 stars [23].

While the development tools and the application of augmented reality are becoming more abundant, AR specific testing tools and practices are lagging behind. One famous virtual reality headset, the "Sword of Damocles", was invented in 1968 by Ivan Sutherland [24]. Yet, while the VR technology has existed for a longer period of time than AR, there are no common test automation practices for application interactions in VR [25]. Figure 3 gives the span of degree of similarity and relation between AR and VR [26].

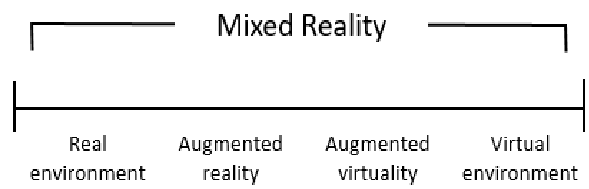

Figure 3. The Real-Virtual Continuum

While previous research has identified a potential void in testing practices for novel interfaces, that is not to say that industry users won't search them out, or rely upon them. Test automation solutions for web and mobile are available from both open source communities and commercial vendors. Test automation tools specifically for AR apps are not as easily found. Figure 4 identifies Gartner survey respondents answers to a question about technology adoption specifically in support enterprise multiexperience app development. Gartner analysts in their analysis of the survey have stated, "The respondents had fairly equal adoption rates across the technology categories we asked about, but Gartner advises application leaders to emphasize adoption of test automation tools and DevOps practices." [5].

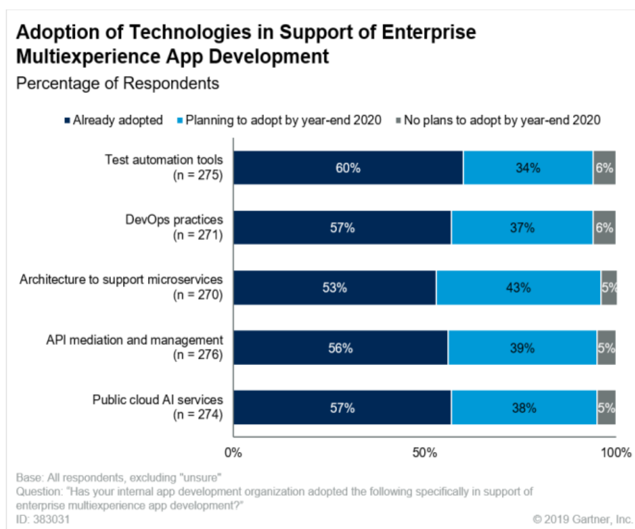

Figure 4. Gartner Survey Analysis: Insights to Kick-Start an Enterprise Multiexperience Development Strategy Findings on Adoption of Technologies in Support of Enterprise Multiexperience App Development

Assessing the quality of an AR application is more complex than the quality of a typical traditional software product. Early adopters of mobile AR applications have experienced bugs in production versions. Reports of quality issues found in Amazon's application ARView include failure to land models on shiny physical surfaces, low model detail, unnatural and distinguishable model outlines, and failure to proportional size a model to the physical environment [27]. Ikea's AR application has had similar quality issues, mostly within low-light conditions, where models would float or drift away from the original anchor point [28].

Evaluating AR systems will be problematic because the technology is young and we do not yet understand the expectations of end-users of AR apps [29]. In addition to those issues, traditional usability evaluation methods are not comprehensively adequate for detecting defects in applications having novel interfaces of which the evaluation methods were designed for and matured over time [30]. Nielsen defined usability as software attributes of learnability, efficiency, memorability, satisfaction, and error [31]. AR usability requires both the physical and virtual components to work together. Thus if either is degraded, such as a low-light physical scenario, the AR application quickly loses usability. There is also the concern of color sameness and collision between models and physical environments. Presence and collaboration in an augmented environment are also new quality concerns compared to traditional testing procedures [32]. The application must be intelligent in these cases to instruct the user to improve the environmental settings or move to a more traditional application interface to solve their needs. This implication of an augmented reality application being intelligent to understand a degraded physical environment's impact to its own usability means quality concerns must be addressed at design.

Here an approach is proposed that produces a quality model to meet the growing expectations from the AR products. This model uses features from the ISO 25010 model.

II. SOFTWARE QUALITY MODEL OF AUGMENTED REALITY APPLICATIONS

A well-designed augmented reality application blends virtual and physical worlds into an experience that facilitates perspective, presence, interaction, and immersion [33]. Perspective is necessary for engaging virtual objects realistically within dynamic physical environments. Presence is the user's ability to be within their physical environment, and engage in the virtual model or scene. This engagement is the interaction that occurs, as we manipulate the model, or as it changes in relation to the perspective of our environment. This all works to create *immersion*, our ability to be in both physical and virtual presence at the same time. However, to scale AR, the application must also have persistence. This allows the augmented models and scenes to be experienced by other application users, scaling the augmented reality experience. These characteristics of augmented reality applications require the developers to go beyond traditional testing concerns.

A. Environmental Context as it Pertains to Functionality

An augmented reality application's functionality must be tested for two distinct requirements, occlusion and collision [34], as discussed below. Users of AR applications need realistic interactions between the physical and virtual objects. This means objects having proper shadows, depth and they should not occupy the same space at the same time.

Occlusion requires that if a virtual object is distanced away from physical objects, during rendering the physical objects should hide or *occlude* portions or the virtual object entirely in a realistic manner. Occlusion is difficult because AR engines must properly represent the physical world, even as our relation to it changes based upon physical movement. The AR engine must reconstruct the 3D model according to the properties of the physical world, and render the dynamic user interface appropriately. Test cases must be created that mimic the possible multitudes of physical realms the application will run in, and then load and engage the virtual objects to test propensity to fail occlusion or create an experience that does not mimic reality. Figure 8 represents failed occlusion and the existence of collision.

A *collision* occurs when a virtual object approaches a physical object and attempts to occupy the same 3D space at a time. Many AR applications will allow the virtual object to be manipulated as it appears in the physical plane. However, the virtual object should not occupy the same space as the physical objects, despite the manipulation. Collision detection needs to be tested in an implementation. Test cases should also include how color and light change the ability to detect and prevent collision.

B. Environmental Context as it Pertains to Visual Distinction

AR applications can lose visual distinction based upon the physical world they are operated within. Physical environments that make gradient distinction difficult, will reduce the usability of the applications. Engaging virtual objects is difficult and prone to fail in areas of intense light, where the color white is dominant, when the model and the physical scene seem chameleon in color, or when surfaces are smooth and reflective in general. Distinction testing must have multiple test cases that alter the physical environment to these types of extremes, to understand the application's behavior, and to potentially present the user with options outside of the AR user interface, such as a traditional menu based interface, in the case failure to render virtual objects such that they are distinct.

C. Environmental Context as it Pertains to Portability

Another environmental concern to AR applications are the variations among the platforms (runtime OS and the computational and display/interaction hardware). There are many physical form factors, and versions of device operating systems, thus an AR application will need to be tested for correct execution across devices and platforms. This is where utilizing a device farm, as we would for typical web and mobile development, will help assure the breadth of device compatibility needed. For those utilizing AR applications internally to an organization, and having a policy on acceptable hardware, the concern for variations in the mobile runtime platform may be diminished. There are many mobile device farm vendors to select from, some having both on premise and cloud options. Example vendors having both on premise and cloud offerings include Experitest, Mobile Labs and SmartBear, as well as others. Crowd testing could also be utilized to establish a large test bed of diverse devices. Crowd testing can be facilitated internal or external to an organization, through crowd testing platforms. Crowd testing can accomplish quality goals such as user experience, functional or regression testing, and it offers diversity beyond devices, but also in geography and in the skills and experience of end users. Crowd testing platforms, as examples, could include RainforestQA, Applause or testIO.

D. Performance Engineering

Three-dimensional rendering while operating camera hardware and merging the image into a user interface is computationally expensive. Early releases of Google's ARCore came with release notes that suggests that the anchors, (a type of feature point that can represents the virtual objects presence in the physical world), should be detached or deleted to prevent expensive CPU costs [35].

Augmented reality applications are often implemented using a distributed architecture, using both local client and some cloud or centralized services. In these cases, we must do performance analysis on the centralized resources, with the understanding that the computational capacities of those assets may be stretched when many clients engage the services. The client devices themselves may also have performance concerns, including battery depletion over long runtimes. Traditional performance testing will solve much of this analysis, and those utilizing cloud resources will need to add validation of the complexity and cost of automatically scaled resources to test plans. Cloud performance can be degraded when there is high network traffic to the cloud's data center or even temperature within the data center [36]. The traffic could be heavy in a distributed AR application implementation.

E. Scaling Usability with Persistence

Scaling augmented reality experiences across multiple users requires centralizing physical maps and virtual object anchors so that multiple users can share immersion. ARKit and ARCore both have these capabilities [37]. Persistence plays into quality for single user scenarios, you place the virtual object and days later expect to see it embedded into the same physical location. Persistence is also a requirement of scaling users' ability to share augmented experience. Test cases will require an amount of complexity in the numbers of testers that engage in one shared scenario and experience, as well as the duration of time the test case covers.

F. Creating the Quality Model

Test planning for an augmented reality application release has many environmentally contextual factors. The virtual object needs to behave within acceptable limits of the physical world. The device runtime environment and form factor are variables as well. Depending upon the nature of the application, cloud technologies such as shared anchor points may be required, which can also lead to having more complex functional and non-functional quality requirements.

Table II attempts to modernize aspects of the ISO 25010 model to reflect the user experience expectations, broken down into the quality characteristics of perspective, presence, interaction, immersion, persistence, and performance. User experience itself is a very influential quality attribute [38]. This model attributes eight characteristics to the user experience of an augmented reality application. One previous research has indicated three characteristics of user involvement, interaction between user, product and other agents, and what can be observed or measured [39]. While another work found five characteristics of aesthetics, appeal, joy, usability, and utility [40]. The model is meant to reflect the ISO 25010 model and relate to the original eight quality characteristics as possible.

TABLE II. Augmented Reality Quality Characteristics

User Experience	**Description**
Functionality / Presence	Occlusion and Collision characteristics will make the AR application more realistic giving more functional uses. This characteristic allows users to be present in both physical and virtual worlds.
Visual distinction / Perspective	The environment, such as highly reflective and bright lighting may affect visual distinction of the application's ability to inject virtual into physical consistently. Perspective is assured by testing for realistic rendering of virtual objects in relation to their physical counterparts so that height, width, and depth are accurately maintained.

Performance / Interaction	Multiple anchors achieved by having many virtual objects in one physical layout can impact device performance. This can limit the user's ability to interact if interaction requires frequent or multiple virtual object creations, instances or mutations. Performance also depends upon cloud anchor mechanisms so that multiple users can share augmented reality experiences.
Portability / Immersion	The many mobile form factors and operating systems create a complex runtime environment to be tested cross-device. For many users of an augmented reality application to experience immersion, the application must be compatible across many device types and versions.
Usability / Persistence	Scaling to achieve shared augmented reality experience across multiple simultaneous users requires persistence. Persistence utilizes cloud technologies to enable distributed users to share anchor points and other object and scene metadata so that they can become immersed in each other's virtual and physical worlds.

G. Applying Automated Testing to AR Applications using Machine Learning

Automation helps accelerate testing so that we can consider production release sooner and also so that we may find more bugs in a shorter amount of time. Similar to how traditional test practices do not consider novel user interfaces such as AR, testing tools are also lacking in this area. One case study by SmartBear is available to discuss their approach [41]. While they describe the physical mobile lab and physical robotics that reposition the devices allowing their cameras to adequately zoom and reset for incorporating the 3D model into the physical world, the case study does not portray how they tested the functionality and user experience as described in our quality model, or specifically how they tested for aspects such as collision or occlusion.

We present a framework that includes test automation, common tools and practices, and the use of machine learning to build an approach to testing that provides probability of accurate model presentation in the physical world, to accelerate manual testers and developers so that they may quickly hone in on problematic areas.

This approach begins with the model files being utilized within the AR development. Figure 5 is taken from an integrated development environment, Android Studio, and illustrates one example of how model files can be stored and organized during AR application development. These become a source of comparison to be utilized in the proposed automation framework.

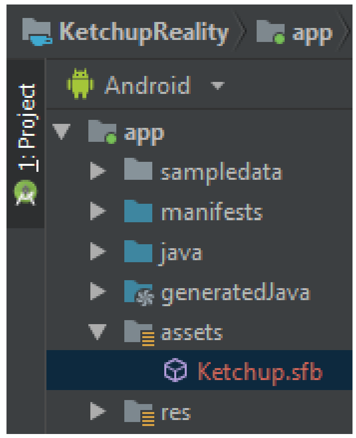

Figure 5 Android Studio Folders and sfb Model File

What is important about these models is the return value when they are sent into common ML image recognition engines. Our framework uses a common open source test automation tool, Appium, to take screenshots during the application under test runtime.

This suggested framework has access to the model within the development environment, as well as the screenshot of the model when utilized within the mobile app. Figure 6 is an image of the model within the development environment. Figure 8 is a screenshot of an AR app running where the model is now occluding. The two images are sent to an ML driven image recognition engine. When the two resultants are within an acceptable value over time, we have accelerated through the testing effort by assuring the model is being detected with confidence when being displayed in the physical world. This is one aspect in how machine learning and test automation can assist in the entire software development lifecycle of augmented reality applications. The same type of harness can be created and run autonomously while the application is in production, again assuring that within the changing environments, the models are being adequately presented in the novel user interface of AR.

As a direct example of this framework, we can submit the model of the ketchup bottle as pictured in figure 6 to the AWS Rekognition service. As indicated in figure 7, the service returns a probability of 97.6% certainty that it is both

Food as classification and it is identifiable as a Ketchup bottle.

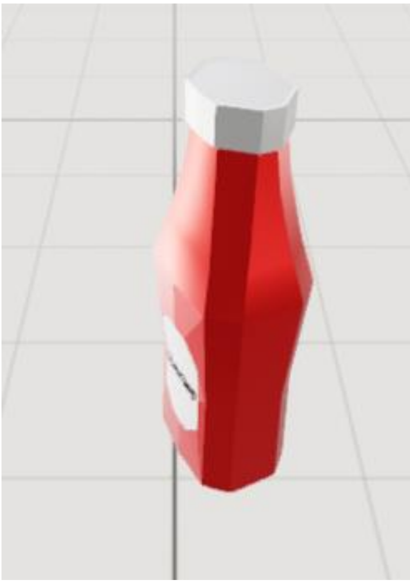

Figure 6. Model file of Ketchup Bottle


```
"Labels": [
    {
        "Name": "Ketchup",
        "Confidence": 97.69482421875,
        "Instances": [],
        "Parents": [
            {
                "Name": "Food"
            }
        ]
    },
    {
        "Name": "Food",
        "Confidence": 97.69482421875,
        "Instances": [],
        "Parents": []
    }
```


Figure 7. Return values from Rekognition Service for Figure 6

However, when we utilize a screenshot of the same model file utilized in an AR app, we see occlusion failing and the ML service returns concerning results. This would indicate potential defects and manual testing could then hone in on this particular test case.

Figure 8 indicates what failed occlusion looks like, and figure 9 gives the Rekognition return types that would help indicate a defect. The bottle is anchored to the back corner of the room, yet is also floating and occluding the guitar that is nearest the mobile device. The probabilities from the same AWS recognition engine have changed dramatically. We no longer see a bottle or food. Rather we have detected Furniture, Leisure Activities, and a Guitar.

Figure 8. Screenshot taken of mobile AR app


```
"Labels": [
    {
        "Name": "Furniture",
        "Confidence": 99.3681869506836,
        "Instances": [],
        "Parents": []
    },
    {
        "Name": "Leisure Activities",
        "Confidence": 88.02964782714844,
        "Instances": [],
        "Parents": []
    },
    {
        "Name": "Guitar",
        "Confidence": 74.00513458251953,
        "Instances": [
            {
                "BoundingBox": {
                    "Width": 0.5425460338592529,
                    "Height": 0.39460861682891846,
                    "Left": 0.30224451422691345,
                    "Top": 0.5120397210121155
                },
                "Confidence": 74.00513458251953
            }
        ],
        "Parents": [
            {
                "Name": "Leisure Activities"
            },
            {
                "Name": "Musical Instrument"
            }
        ]
    },
```


Figure 9. Return Values from Rekognition Engine for Figure 8

III. Conclusions

Augmented reality applications merge virtual models and the physical world. These applications are becoming more popular in many verticals, and several SDKs are available to assist in creation. However further development of testing methodologies and tooling is needed. The novel features of an AR user interface is not adequately addressed by traditional testing methods, or of the ISO 25010 model. This research suggests creating test cases that focus on characteristics of perspective, presence, interaction, immersion, persistence, and performance. Automation and machine learning of image detection features also need to be leveraged to assist in the detection of potential defects in the AR applications. The revised quality model and the ML enabled automation framework seek to expand current capabilities and methods to enhance defect detection in AR applications.

REFERENCES

[1] X. Wang, A. Kotranza, J. Quarles, B. Lok, B.D. Allen, A pipeline for rapidly incorporating real objects into a mixed environment, in The 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, Washington, DC, USA, October 2005, pp. 170-173.

[2] I. Rabbi, S. Ullah, A survey on augmented reality challenges and tracking, Acta Graphica: J. Print. Sci. Graphic Commun. Volume 24, No. 1-2, 2013, pp. 29-46.

[3] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," Behaviour and Information Technology, vol. 22, pp. 397-409, 2003.

[4] Gartner Emerging Technology Analysis: Augmented and Mixed Reality Opportunity for 3D Design Software and Vertical ISVs, Tuong Nguyen, 23 August 2019.

[5] Gartner Survey Analysis: Insights to Kick-Start an Enterprise Multiexperience Development Strategy, Jason Wong, Adrian Leow, 12 February 2019.

[6] "Google-ar/arcore-android-sdk" GitHub. February 23, 2018. Accessed April 28, 2019. https://github.com/google-ar/arcore-android-sdk

[7] Apple's ARKit: Cheat sheet, Cory Bohon, TechRepublic, June 7, 2019. Accessed September 28, 2019. https://www.techrepublic.com/article/apples-arkit-everything-the-pros-need-to-know/

[8] [8] Facebook rebrands Camera Effects Platform, expands program to Instagram, Robert Williams, October 8, 2018. Accessed September 28, 2019.

https://www.mobilemarketer.com/news/facebook-rebrands-camera-effects-platform-expands-program-to-instagram/539066/

[9] Announcing the 1.0 Release of the QCAR SDK, April 26, 2011. Accessed September 28, 2019. https://developer.vuforia.com/forum/news-and-announcements/announcing-10-release-qcar-sdk

[10] Qualcomm Sells Its Vuforia Augmented Reality Business to PTC for $65 Million, Ina Fried, October 12, 2015. Accessed September 28, 2019. https://www.vox.com/2015/10/12/11619500/qualcomm-sells-its-vuforia-augmented-reality-business-to-ptc-for-65

[11] See the Future Create the Augmented World, 2019. Accessed September 28, 2019. http://maxst.com/#/en/aboutus

[12] Release Notes Wikitude Sdk, 2012. Accessed September 28, 2019. https://www.wikitude.com/external/doc/documentation/latest/android/changelog.html

[13] Y. Arifin, T. G. Sastria, and E. Barlian. "User experience metric for augmented reality application: a review." Procedia Computer Science 135 (August 29, 2018): 648–56.

[14] Use Vuforia for computer vision and augmented reality apps, 2019. Accessed September 28, 2019. https://www.ptc.com/en/academic-program/academic-products/free-software/vuforia

[15] Downloads, 2019. Accessed September 28, 2019. https://developers.facebook.com/docs/ar-studio/downloads/

[16] Get ready for the latest advances in augmented reality, 2019 Accessed September 28, 2019. https://developer.apple.com/augmented-reality/

[17] ARCore overview, February 28, 2019. Accessed September 28, 2019. https://developers.google.com/ar/discover/

[18] Wikitude Augmented Reality SDK, 2019. Accessed September 28, 2019. https://www.wikitude.com/products/wikitude-sdk/

[19] AR SDK, 2019. Accessed September 28, 2019. http://maxst.com/#/en/arsdk

[20] S.P. Yun, H.J. Yap, R. Singh, S.W. Chang, K.L.R. Cheong, Augmented reality assisted factory layout planning and analysis for a flexible manufacturing cell, in International Conference on Computer Science and Computational Mathematics, 2014, pp. 106-111.

[21] S. J. Henderson, S. Feiner, Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret, in 8th IEEE International Symposium on Mixed and Augmented Reality, Orlando, FL, USA, 2009, pp. 135-144.

[22] G. Westerfield, A. Mitrovic, M. Billinghurst, Intelligent augmented reality training for motherboard assembly, Int. J. Artif. Intell. Educ. Volume 25, No. 1, 2015, pp. 157-172.

[23] Mercedes cAR app. Car parking in the living room, 2018. Accessed September 28, 2019. https://developer.mercedes-benz.com/inspire/cARapp

[24] Y. A. G. V. Boas, "Overview of virtual reality technologies." In Interactive Multimedia Conference, vol. 2013.

[25] A. Bierbaum, P. Hartling, Patrick and C. Cruz-Neira, "Automated testing of virtual reality application interfaces. Eurographics Workshop on Virtual Environments," pages 107–114. ACM Press, New York, 2003

[26] R. T.Azuma, Y.Baillot, R.Behringer, S. K. Feiner, S. Julier, and B. Macintyre, Recent advances in augmented reality. Computer Graphics and Applications 21, 6 (Nov./Dec. 2001), pp. 34–47.

[27] Amazon's AR Tool is a useful way to see products before you buy them, Phillip Tracy, November 2, 2017. Accessed September 28, 2019. https://www.dailydot.com/debug/amazon-ar-view-app/

[28] J. Diaz, Avoid The Dreaded Ikea Fight With This New App, , September 22, 2017. Accessed September 28, 2019. , https://www.fastcompany.com/90143908/avoid-killing-your-relationship-at-ikea-with-this-new-app

[29] A.Dünser and M Billinghurst "Evaluating augmented reality systems," In: Furht B. (eds) Handbook of Augmented Reality. Springer, New York, NY, 2011, pp 289-307.

[30] A. Sutcliffe and K. Kaur, "Evaluating the usability of virtual reality user interfaces," Behaviour and Information Technology, vol. 19, 2001.

[31] J. Nielsen, Usability Engineering. San Francisco: Morgan Kaufmann, 1993.

[32] K. M. Stanney, M. Mollaghasemi, L. Reeves, R. Breaux, and D. A. Graeber, "Usability engineering of virtual environments (VEs): identifying multiple criteria that drive effective VE system design," Int. J. Hum.-Comput. Stud., vol. 58, pp. 447-481, 2003.

[33] F. Tao, M. Zhang, and A.Y.C. Nee. Digital Twin driven smart manufacturing. Elsevier Academic Press, 2019. pp. 229.

[34] D. E. Breen, E. Rose, and R. T. Whitaker. "Interactive occlusion and collision of real and virtual objects in augmented reality." Technical report ECRC-95-02, European Computer Industry Research Center, Munich, Germany, 1995.

[35] "Fundamental Concepts." Google Inc. ARC Core, February 28, 2019. Accessed April 28, 2019.

https://developers.google.com/ar/discover/concepts#anchors_and_trackables.

[36] A. N. Asadi, M. A. Azgomia, and R. Entezari-Malekib. "Unified power and performance analysis of cloud computing infrastructure using stochastic rewardn." Computer Communications, Volume 138, 15 April 2019, pp.67-80.

[37] "Persistence in AR - Why Is It so Important?" AR Critic. June 16, 2018. Accessed May 01, 2019. https://arcritic.com/2151/persistence-in-ar-why-is-it-so-important/

[38] J. York and P. C. Pendharkar, "Human–computer interaction issues for mobile computing in a variable work context," International Journal of Human- Computer Studies, vol. 60, pp. 771-797, 2004.

[39] W. Albert, T. Tullis Measuring the user experience: collecting, analyzing, and presenting usability metrics. Newnes; 2013 May 23.

[40] M. Hassenzahl and N. Tractinsky, "User experience - a research agenda," Behaviour & Information Technology, vol. 25, pp. 91-97, 2006.

[41] J. Kaasila, "Google Augmented Reality - How Bitbar Helped Automate Their App Testing." Bitbar, April 17, 2018. Accessed September 28, 2019. https://bitbar.com/blog/how-bitbar-helped-google-to-automate-their-augmented-reality-testing/.