INVITED PAPER

Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights

In the context of mobile augmented reality, this article addresses a challenging network scenario that requires adaptation in the usage of computing, storage, and communication resources.

By Xiuquan Qiao, Pei Ren, Schahram Dustdar, *Fellow IEEE*, Ling Liu, *Fellow IEEE*, Huadong Ma, and Junliang Chen

ABSTRACT | Mobile augmented reality (Mobile AR) is gaining increasing attention from both academia and industry. Hardware-based Mobile AR and App-based Mobile AR are the two dominant platforms for Mobile AR applications. However, hardware-based Mobile AR implementation is known to be costly and lacks flexibility, while the App-based one requires additional downloading and installation in advance and is inconvenient for cross-platform deployment. In comparison, Web-based AR (Web AR) implementation can provide a pervasive Mobile AR experience to users thanks to the many successful deployments of the Web as a lightweight and cross-platform service provisioning platform. Furthermore, the emergence of 5G mobile communication networks has the potential to enhance the communication efficiency of Mobile AR dense computing in the Web-based approach. We conjecture that Web AR will deliver an innovative technology to enrich our ways of interacting with the physical (and cyber) world around us. This paper reviews the state-of-the-art technology and existing implementations of Mobile AR, as well as enabling technologies and challenges when AR meets the Web. Furthermore, we elaborate on the different potential Web AR provisioning approaches, especially the adaptive and scalable collaborative distributed solution which adopts the osmotic computing paradigm to provide Web AR services. We conclude this paper with the discussions of open challenges and research directions under current 3G/4G networks and the future 5G networks. We hope that this paper will help researchers and developers to gain a better understanding of the state of the research and development in Web AR and at the same time stimulate more research interest and effort on delivering life-enriching Web AR experiences to the fast-growing mobile and wireless business and consumer industry of the 21st century.

KEYWORDS | 5G; augmented reality (AR); cloud computing; edge computing; mixed reality; mobile augmented reality (Mobile AR); osmotic computing; virtual reality (VR); Web-based augmented reality (Web AR).

Manuscript received May 1, 2018; revised December 14, 2018; accepted January 17, 2019. Date of publication February 18, 2019; date of current version March 25, 2019. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61671081, in part by the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007, in part by the 111 Project under Grant B18008, in part by the Beijing Natural Science Foundation under Grant 4172042, in part by the Fundamental Research Funds for the Central Universities under Grant 2018XKJC01, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2019213. The work of L. Liu was supported in part by the National Science Foundation under Grant NSF 1547102 and Grant SaTC 1564097 and in part by the IBM Faculty Award.

(Corresponding authors: Xiuquan Qiao; Schahram Dustdar.)

X. Qiao, **P. Ren**, **H. Ma**, and **J. Chen** are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: qiaoxq@bupt.edu.cn; renpei@bupt.edu.cn; mhd@bupt.edu.cn; chjl@bupt.edu.cn).

S. Dustdar is with the Distributed Systems Group, Technische Universität Wien, 1040 Vienna, Austria (e-mail: dustdar@dsg.tuwien.ac.at).

L. Liu is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ling.liu@cc.gatech.edu).

Digital Object Identifier 10.1109/JPROC.2019.2895105

I. INTRODUCTION

The phenomenal growth of augmented reality (AR) [1]–[3] over the past decade has attracted significant research and development efforts from both academia and industry. By seamlessly integrating virtual contents with the real world, AR makes it possible to provide users with

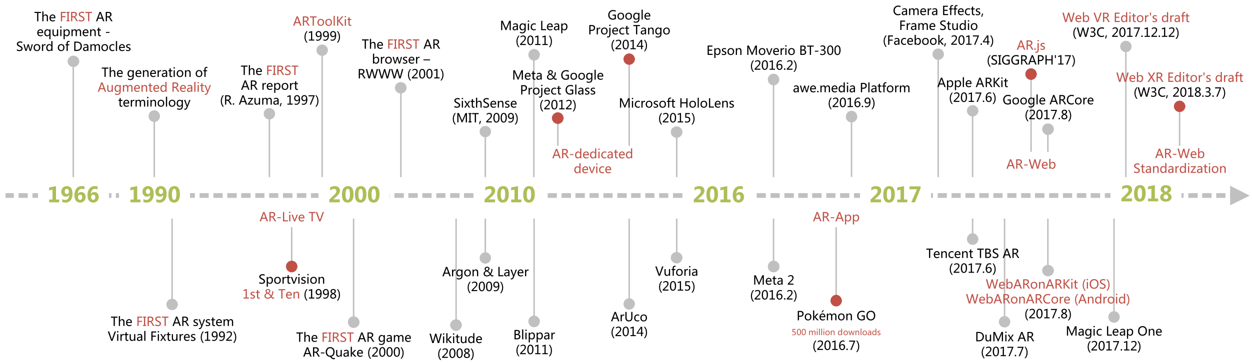


Fig. 1. ***Historical evolution of AR.***

a sensory experience beyond reality. Especially, in recent years, the advances in the following three technologies have further fueled the research and development of AR: the emergence of dedicated AR devices (e.g., Google Glass, Microsoft Hololens and Epson Moverio BT-300,[1] and Magic Leap) and powerful development kits (e.g., ARCore and ARKit), the improvements in the performance of mobile devices and sensor integration, and advances in computer vision (CV) technologies. AR has offered tangible benefits in many areas, such as entertainment, advertisement, education, navigation, maintenance, and so on [4]–[8]. For example, Pokémon GO, a location-based AR game, has reached over 500 million downloads in more than 100 countries within just eight weeks of its public release [9]. Both AR and virtual reality (VR) can alter the perception of our presence in the world. However, AR, unlike VR, which transposes our presence in the world to elsewhere, allows users to be present in the world and simply "augments" our perception of the world by adding the ability to provide users with contextually relevant information beyond our current perceived state of presence. Digi-Capital [10] forecasted that the global VR/AR market would reach 108 billion dollars by 2021, and Cisco [11] reported that the global AR traffic would increase sevenfold between 2016 and 2021.

The historical evolution of AR is shown in Fig. 1. Beginning from the first time Sportvision applied AR to live TV (1st & Ten, 1998), and then with the first dedicated AR device (Google Glass, 2012) and smartphones (Project Tango, 2014), and on to the first phenomenal AR App (Pokémon GO, 2016), it has become clear that both AR technologies and devices tend to be powerful, mobile, and lightweight. However, the current mobile augmented reality (Mobile AR) hardware and operating systems (e.g., Embedded Linux, Android, iOS) present a complex diversity. Most Mobile AR applications or solutions are designed based on a specific platform and lack cross-platform support. To reach more users, an AR application needs to go through repeated development cycles to accommodate different platforms [12], which undoubtedly increases the cost of development and deployment. Although there are already some preliminary attempts toward Web-based AR (Web AR), the limited networking and computing capability greatly hinder its practical application. Since 2017, the Web AR provisioning solution has gradually attracted developers' attention again due to the ever-increasing development of user device and mobile network and has emerged as a promising direction for Mobile AR.

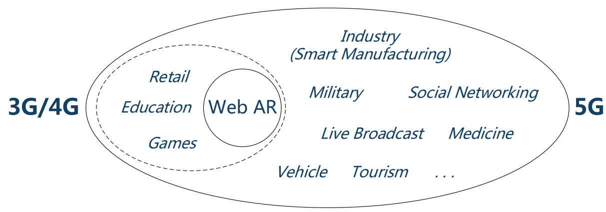


Fig. 2. ***Current and future application areas of Web AR.***

The invention of the World Wide Web marked the beginning of a new era, with a Web-based service provisioning paradigm. The native cross-platform and lightweight features of the Web simplify service access for users, thereby facilitating the large-scale promotion of Web-based applications. Besides Web browsers, many mobile Apps (e.g., Facebook and Snapchat) nowadays are also designed in a hybrid (Native + Web) way, which has both the advantages of good interaction experience and cross-platform support. All of these infrastructures provide a common platform for the pervasive promotion of Web AR. Here, we define Web AR as a type of Web AR implementation approach.

Although the technology of the Web offers a promising approach for the cross-platform, lightweight, and pervasive service provisioning of Mobile AR, there are still several challenges to applying Web AR in real cases. Computational efficiency, energy efficiency, and networking are three important challenges. AR is a computation- and data-intensive application. The limited computing and rendering capabilities on the Web make it more challenging to achieve a high-performance and energy-efficient Web AR. First, the limited performance of a Web AR application will significantly degrade the user's experience. Second, the battery on the mobile device will face tremendous

[1]Microsoft Hololens and Epson Moverio BT-300 received the Red Dot Award (Product Design category) in 2016 and 2017, respectively.

pressure caused by the complex computation tasks, as it is only designed for common functionalities. To achieve better performance, Web AR applications usually take advantage of a way to off-load computation (e.g., cloud computing) to accelerate the process. However, computation offloading may introduce an additional communication delay, which will impact the user experience and limit its application under the current mobile networks.

The good news is that several technological advances have started to enter the landscape of Mobile AR. First, the upcoming 5G networks [13] bring new opportunities for Mobile AR, especially Web AR. They provide higher bandwidth (0.1∼1 Gb/s) and lower network delay (1∼10 ms), which improves the data transmission on mobile networks. Second, the introduction of new characteristics, such as multiaccess edge computing[2] (MEC), device-to-device (D2D) communication, and network slicing, provides an adaptive and scalable communication mechanism that further provides efficient infrastructures for the deployment and promotion of Web AR. The soon-to-be-available 5G networks and the rapid performance improvement of mobile devices, therefore, have laid a solid foundation for the practical deployment and application of Web AR on a large scale.

The rest of this paper is organized as follows. Section II presents the principles of Mobile AR and three typical implementation mechanisms, as well as the challenges and enabling technologies for when AR meets the Web. Section III summarizes the different Web AR implementation approaches based on experience from real-world use cases and experiments. Section IV discusses the challenges ahead and some future research directions. We conclude this paper in Section V.

II. BACKGROUND: MOBILE AR PRINCIPLES AND TYPICAL IMPLEMENTATION MECHANISMS

AR was defined as a technology that integrates virtual objects with the 3-D real environment in real time and supports interaction by Azuma [2]. In this section, we give an overview of the principles of Mobile AR and summarize the three typical Mobile AR implementation mechanisms, followed by the challenges the Web AR will face when applying it to real cases. Finally, we detail some enabling Web technologies that are necessary or recommended for the implementation of Web AR.

A. Mobile AR Principles

AR is a visual technology between VR and real reality. By superimposing computer-generated virtual content over the real world, AR can easily help users to better understand their ambient environment.

[2]MEC (September 2017) [14], formerly mobile edge computing (September 2015) [15], refers to the enabling technologies that provide computing capabilities and service environment at the edge of the network [European Telecommunications Standards Institute (ETSI) White Paper].

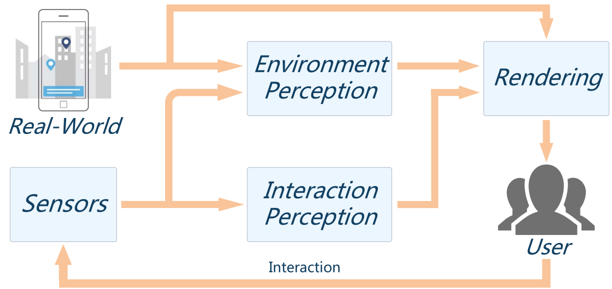


Fig. 3. ***Typical AR process.***

A typical AR process is shown in Fig. 3. The camera and other types of sensors are used to continually gather user ambient information. The environment perception analyzes the captured information (e.g., image/video, location, and orientation) for real-world recognition and perception. In the meantime, the user's interaction information is also gathered by the sensors and then analyzed for the purpose of tracking objects. Both the results of the perception of the environment and the interaction are used for a seamless integration of virtual contents with the real world, i.e., a rendering operation is performed, after which the AR will be presented to the user.

B. Typical Implementation Mechanisms

The advances in mobile devices, including computing and display platforms, provide more choices for the implementation of AR applications. In accordance with the tracking technologies, we detail the typical Mobile AR implementation mechanisms in terms of three aspects, that is, sensor-based, vision-based, and hybrid tracking methods.

Different implementation mechanisms are naturally with different complexities regarding computing, networking, and storage. The sensor-based method is a relatively lightweight Mobile AR implementation approach, while in contrast, the vision-based approach places high demands on the computing and storage capabilities of the runtime platform, as well as network capability. As shown in Fig. 4, the hybrid tracking mechanism is obviously a compromise solution.

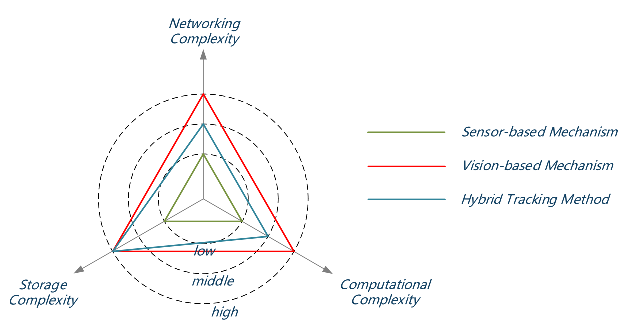


Fig. 4. ***Computational/storage/networking complexities for the three typical implementation mechanisms.***

1) Sensor-Based Mechanisms: Mobile devices nowadays already support a variety of sensors, such as accelerometers, gyroscopes, compasses, magnetometers, GPS, and so on. Much effort has been devoted to this type of implementation mechanism [16]–[18]. An obvious example of this is Pokémon GO, which provides AR experience by leveraging the location-based service technology, has launched an unprecedented revolution in the field of mobile AR. Note that the camera can be enabled to capture the surrounding environment, but only for the display of the environment as the background. In addition to Mobile AR implementation mechanisms based on a single sensor, combining different sensors allows many applications to achieve more accurate tracking results [19]–[21]. The increasing of sensor category, as well as the continuous enhancement of sensor functionality, provides the basis and opportunities for the diversification of Web AR applications. Considering the complexity of computation, storage, and networking, this lightweight Web AR implementation mechanism is currently the lowest option for users to get started. However, this method works in an open-loop way, which will result in an unavoidable cumulative error, since the tracking error cannot be evaluated and corrected in real time.

2) Vision-Based Mechanisms: Similarly, the camera on the device captures the surrounding environment, but it further provides the basis for vision-based object recognition, detection, and tracking. This type of mechanism uses feature correspondences to estimate pose information to align the virtual content with real-world objects and is analogous to a closed-loop system. Depending on different features, it can be divided into two methods, as discussed in the following. The frame-by-frame tracking approach avoids the aforementioned error accumulation. However, it introduces heavy computational pressure on mobile devices, especially for natural feature tracking methods. Besides the improvement of device capability, the advances in network (e.g., the upcoming 5G networks) will provide another approach to the problem of inefficient Web AR application performance, i.e., computation outsourcing (see Section III-B).

The *marker-based method* uses a predefined marker to meet the tracking requirement, including two ways as follows.

1) The fiducial method has predefined shape, size, color, and properties, as shown in Fig. 5. It can achieve superior accuracy and robustness in changing environmental conditions. Its easily identified features made it popular in the early stages of development [23]–[25]. However, the difficulty of deploying and maintaining fiducials in an unknown or outdoor environment has limited the scope of its application.
2) The natural feature tracking method avoids the difficulties of the fiducial tracking method mentioned earlier and thus has a broader range of applications.

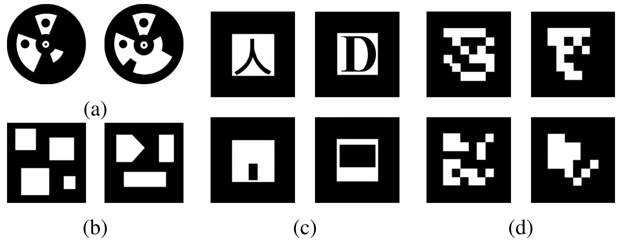


Fig. 5. ***Several planar pattern marker systems [22] used in AR. (a) Intersense. (b) ARSTudio. (c) ARToolKit. (d) ARTag.***

The two approaches are listed in the following.

1) *2-D Image:* The real-world image method (e.g., based on a photograph or a poster) is an alternative to the fiducial-based method. Note that this method requires a more powerful object detection and recognition algorithm; not all 2-D images can be used for AR pose estimation, for example, a solid color image without any pattern.
2) *3-D Object:* It is natural to extend the tracking of objects from a 2-D image to a 3-D object. Many algorithms are already available for specific nonregular objects, such as human faces, but this is still challenging for general recognition. Although this type of method is in its early stages, its potential value still merits attention.

The *markerless method* detects and understands an unknown or outdoor real-world environment (e.g., the locations of walls), and no preknowledge of the environment is required, which will efficiently promote large-scale Mobile AR. However, it is more challenging to adopt simultaneous localization and mapping (SLAM) [26], the core part of markerless environment perception, to Mobile AR, mainly because of the computational inefficiency and limitations of the resources of mobile devices. Current solutions mostly rely on a collaboration between SLAM and other sensors.

3) Hybrid Tracking Mechanism: The hybrid Mobile AR implementation mechanism is a compromise, taking into consideration the computational inefficiency of mobile devices. It overcomes the weaknesses and limitations of the individual methods mentioned earlier by combining different methods. Many applications have demonstrated the suitability of this approach [27]–[29]. It not only provides Mobile AR applications with convincing precise and robust results but also reduces the computational complexity. Considering the limited computing capability and network performance, this hybrid scheme will play an important role in the promotion of Mobile AR on a large scale at present.

C. Challenges When AR Meets the Web

In recent years, with the rapid development of hardware, especially with the emergence of artificial intelligence (AI) chips, the computing capability of mobile devices has been greatly improved, which basically satisfies the intensive computational requirements of Mobile AR applications (i.e., App-based Mobile AR implementations). However, there are still several challenges that cannot be ignored when applying AR applications on the Web in real cases.

1) Limited Computing Capability Versus the Requirement of Powerful Computing: Tracking and registration are the two core parts of the AR system and also the most computationally intensive parts. The fiducial tracking method provides an accurate and robust tracking approach for Web AR applications since only simple matrix operations are required. The natural feature tracking method has a broader range of applications than the fiducial tracking method. However, the inefficient computing capability on the Web makes it hard to apply it in real cases. For example, ORB [30], as a lightweight CV algorithm, still cannot meet the computing requirements of AR on the Web. For the markerless Mobile AR implementation method, although there have already been some efforts to reduce the computational complexity of SLAM (e.g., ORB-SLAM [31]), it is still challenging to port it to the Web. In addition to the difficulty of computing on the Web, rendering is another challenge for Web AR, due to the limited rendering capability of Web browsers. Moreover, diverse and inefficient computing platforms (e.g., built-in browser) also result in a degradation of the performance of Web AR. As mentioned earlier, it will be an efficient and promising approach for Web AR by offloading computation-intensive tasks to the edge or remote cloud for acceleration (see Section III-B), especially in the upcoming 5G networks.

2) Network Delay Versus the Requirement of Real-Time Performance: Cloud servers always have a more powerful computing capability and thus provide a performance improvement for Web AR applications. However, AR is a computation- and data-intensive application. Large communication delays are introduced by offloading computing tasks to the cloud. It is therefore difficult for current mobile networks to support real-time operations (e.g., tracking and interaction), due to the limited data rate and unacceptable network delay. Web AR applications are more dependent on mobile networks. The advanced technologies, such as software-defined network and network function virtualization, which provide us opportunities for the adaptive and intelligent network resource scheduling (e.g., network slicing in the 5G era) and further differentiated Web service provisioning according to the application characteristics, are also worth our attention. Besides the computation outsourcing, the self-contained solutions are also important. With the development of the AI chip technologies, computations can be finished at the user equipment, thereby avoiding the extra communication delays caused by network transmissions.

3) Limited Battery Capability Versus Extreme Energy Consumption: Web AR is a power-hungry application, but most Web AR applications nowadays suffer from a limited energy supply. The need for the sensors to cooperate over a long period of time, the analysis of the information, computing, communication, and display, puts tremendous pressure on the battery of the mobile device. However, current batteries on mobile devices are only designed for common functionalities, such as telephone and Internet access. The extreme energy consumption referred to will significantly hinder the deployment of Web AR on common mobile devices. The computation outsourcing mechanism can alleviate the energy consumption of the end device by offloading computing pressure to the cloud, but it also subjects to network conditions.

4) Diverse Enabling Infrastructures Versus the Requirement of Pervasive Promotion: The diversity of computing and display platforms, operating systems, and even data formats gives rise to a serious compatibility challenge. As mentioned earlier, many mobile applications are designed in a hybrid way, where the built-in browsers are simplified for the purpose of being lightweight. However, the diversity of computing platforms hinders the pervasive promotion of Web AR. Moreover, supporting different sensors and display platforms, as well as operating systems, also makes the development of Web AR challenging. Note that the virtual contents created by different tools face compatibility challenges as well for use on the Web. However, all these compatibility challenges require the joint efforts of academia, industry, and standards organizations.

D. Enabling Technologies for Web AR

Some advanced Web technologies nowadays are emerging to meet the basic requirements of Web AR, and, moreover, also provide performance improvement approaches. Fig. 6 shows four major Web-enabling technologies.

1) WebRTC [32]: This technology provides browsers with real-time communications and is one of the most important and basic technologies for Web AR. The camera captures the ambient environment in the form of a video stream by using the WebRTC technology, which provides the basis for further perception of the environment, rendering, and other operations in a Web AR application. A large number of browsers nowadays have already supported this technology. Besides video capture, the WebRTC technology currently also supports video coding, encryption, rendering, processing, and so on. However, considering the limited capability of mobile Web platforms, an efficient WebRTC solution for Web AR is still worth our attention.

2) WebAssembly [33]: To simplify the programming process and achieve native speed, the recently emerged WebAssembly is designed as a computational acceleration approach on the Web by encoding procedures (e.g., C, C++, Rust, and Go) into a size- and load-time-efficient binary format, which can be executed on the

Browsers		Desktop Browser					Mobile Browser					
		Edge	Firefox	Chrome	Safari	Opera	IE Mobile	Firefox for Android	Chrome for Android	iOS Safari	Opera Mobile	Samsung Internet
Enabling Web Technologies for Web AR	WebAssembly	12 – 14 15 16 – 17 18	2 – 46 47 – 51 52 53 – 62 63 64 – 65	4 – 50 51 – 56 57 – 69 70 71 – 73	3.1 – 10.1 11 – 11.1 12 TP	10 – 37 38 – 43 44 – 56 57	10 11	63	70	3.2 – 10.3 11 – 11.4 12.1	12 – 12.1 46	4 – 6.2 7.2
	WebGL	12 – 17 18	2 – 3.6 4 – 23 24 – 62 63 64-65	4 – 7 8 – 32 33 – 69 70 71 – 73	3.1 – 5 5.1 – 7.1 8 – 11.1 12 TP	10 – 11.5 12.1 – 18 19 – 56 57	10 11	63	70	3.2 – 7.1 8 – 11.4 12.1	12 – 12.1 46	4 – 6.2 7.2
	WebRTC	12 – 14 15 – 17 18	2 – 21 22 – 43 44 – 62 63 64-65	4 – 22 23 – 55 56 – 69 70 71 – 73	3.1 – 10.1 11 – 11.1 12 TP	10 – 17 18 – 42 43 – 56 57	10 11	63	70	3.2 – 10.3 11 – 11.4 12.1	12 – 12.1 46	4 – 6.2 7.2
	Web Workers	12 – 17 18	2 – 3 3.5 – 62 63 64 – 65	4 – 69 70 71 – 73	3.1 – 3.2 4 – 11.1 12 TP	10.1 11.5 – 56 57	10 11	63	70	3.2 – 4.3 5 – 11.4 12.1	12 – 12.1 46	4 – 6.2 7.2

= Supported = Partial support = Not supported

Fig. 6. ***Browser support tables of enabling Web technologies (i.e., WebAssembly, WebGL, WebRTC, and Web Workers) for Web AR application up to December 1, 2018 (Source: https://caniuse.com).***

Web directly [34]. Mainstream browsers (e.g., Chrome, Firefox, and Safari) have also started to support this Web technology. WebAssembly solves the bottleneck problem of JavaScript and has therefore caused wide concern. It not only improves Web AR application performance but also makes the development process easier bringing it into a close relation with current mature CV algorithm, for example, OpenCV.js [35], [36], the WebAssembly version of OpenCV. The emergence of WebAssembly will bring a revolution to the Web platform [37].

3) Web Workers [38]: This introduces the multithread technology to JavaScript. It utilizes worker threads to achieve parallelized computing, rendering, and resource loading in an asynchronous way, and moreover, it has already been the part of HTML5 specification. As another computational acceleration approach, Web Workers provide a simple method for program parallelization of Web AR applications, such as 3-D model predownloading and parallelized feature points' matching. By scheduling and balancing the time- and resource-consuming operations in Web AR applications, it can provide users with a better experience, especially under the current mobile networks.

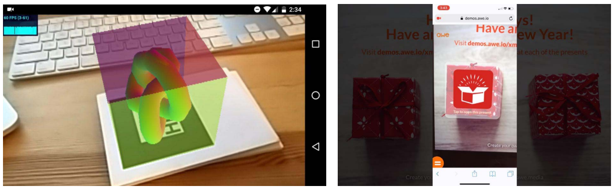
4) WebGL [39]: This provides a hardware-based (GPU) rendering acceleration approach on the Web. Since image processing has a strict requirement of the computing resources, an efficient computing platform is, therefore, important for computation-intensive applications. A set of efficient JavaScript APIs for interactive 2-D and 3-D graphics rendering is available in this library. The use of a GPU in the mobile device makes the presentation of AR smoother and more realistic on the Web. Also, worth mentioning is Three.js [40], a WebGL-based JavaScript library, which helps developers work with 2-D and 3-D graphics on a browser using WebGL in a simpler and more intuitive way. WebGL 2 specification finished in January 2017 and this technology has been widely supported in modern browsers.

The continuous development of these technologies mentioned earlier provides a basis for the Web AR applications and, more generally, will also motivate the innovation of Web-based applications. In the meantime, these applications will further spawn new Web technologies.

III. DIFFERENT WEB AR IMPLEMENTATION APPROACHES

To explore the potential of AR on mobile devices, both academia and industry are now seeking more efficient implementation approaches to compensate for the gap between the user experience of Mobile AR application and the limited capability of the Web browser. Web AR, as a branch of Mobile AR, has recently attracted a great deal of attention due to its lightweight and cross-platform features. Depending on the different computing paradigms, we can classify the Web AR implementation approaches into two types as follows.

1) Self-contained method executes all tasks on the mobile device locally (i.e., offline approach). The advantage of this method is that it is less dependent on mobile networks, so the real-time tracking performance will not be degraded by additional communication delay. However, the inefficient computing capability of the mobile device becomes its fatal flaw; current mobile devices still cannot carry out these tasks very well, especially on the Web.



(a) (b)

Fig. 7. ***State of the art of (a) fiducial-based (AR.js [41]) and (b) natural feature-based (awe.js [42]) Web AR JavaScript library/Plug-in implementations.***

2) Computation outsourcing method leverages the computation and storage capabilities of the cloud servers, and it can usually provide a better user experience than the aforementioned self-contained one. However, this method has a strong dependence on the mobile networks, and therefore, the performance of Web AR applications is easily affected by network conditions.

A. Self-Contained Method

There are two main implementation approaches for the self-contained method. One is to develop pure JavaScript-based libraries or plug-ins to provide Mobile AR services on the Web. The other is to extend the browser kernel to achieve better Web AR application performance. We will now present these two approaches in detail.

1) Pure JavaScript Library/Plug-In: As mentioned earlier, Mobile AR implementation methods based on fiducial tracking can always provide an accurate and robust identifying and tracking performance due to their low computational complexity. Many dedicated JavaScript libraries/plug-ins are already available to support AR services on the Web, including JS-ArUco [43] (a port to JavaScript of the ArUco), JSARToolkit [44] (based on the original ARToolKit [45]), JSARToolKit5 (an emscripten port of ARToolKit), and so on. The state of the art is the newly (in 2017) proposed AR.js [41], a Web AR solution based on Three.js and JSARToolKit5, which can work on all platforms and any browser with WebRTC and WebGL; it achieves even 60 frames/s, stable on Nexus 6P. However, currently, AR.js can only support the fiducial marker, as it involves only simple matrix operations. It is still challenging for AR.js to support natural feature objects. Awe.js [42] is another Web AR implementation, i.e., one based on natural feature tracking (2-D image), and some experimental attempts have demonstrated its suitability. However, the inefficient computing capability of the Web results in significant tracking error due to the complex computational requirements of Web AR applications, not to mention the 3-D object and markerless Web AR implementation methods. Moreover, the aforementioned compatibility challenge regarding different computing capabilities of browsers, including built-in browsers, also makes the pervasive and large-scale promotion of Web AR difficult.

In addition to traditional CV methods [46]–[48], algorithms based on deep learning have also received a lot of attention and development effort in recent years. Web-based neural network algorithms, such as ConvNetJS [49], CaffeJS [50], WebDNN [51], deeplearn.js [52], and TensorFlow.js [53], provide a novel and cross-platform approach for image analysis and processing on the Web. Specifically, these enabling technologies that leverage the convolutional neural networks can be further designed for generalized object detection, recognition, and tracking in a variety of Web AR applications, which provide intelligent context-aware ability and accurate vision-based tracking ability as well, and therefore greatly enhance the capabilities of Web AR. However, the models' size is still an obstacle for its pervasive application. For example, the model size of GoogLeNet [54] in CaffeJS is even up to 28 MB, which is unacceptable for Web users. Moreover, the time of forward pass[3] is also a challenge for the real-time requirements of Mobile AR applications, especially on the Web. The question of model compression and inference acceleration is, therefore, arise [55]–[58] for the purpose of Web AR practical application.

2) Extending the Browser Kernel: The Web browser is nowadays an important entrance for users to connect the Internet. By extending the browser kernel to support AR, Web AR applications can often get near-native performance on mobile devices and thus a better user experience. There have already been some efforts from academia and industry to explore the potential of this Web AR implementation approach, such as RWWW browser [59], Wikitude [60], and the Argon project [61]. The state of the art from Mozilla and Google is Project WebXR Viewer [62], and WebARonARKit and WebARonARCore [63]. These efforts aim to provide a standard environment for Web AR developers. However, they are still in their infancy and have not been applied in practice on a large scale. In addition to Mozilla and Google, there are also other companies making an effort to bridge the gap between the Web and the AR. Both Baidu and Tencent proposed their Web AR solutions in 2017, namely, DuMix AR [64] and TBS AR [65], respectively. Fig. 8 shows the TBS AR system architecture as an example. The browser-kernel extension solution presents a promising and powerful self-contained Web AR implementation solution compared with the pure JavaScript library/plug-in method. However, before the standardization of AR-supported browsers is finished, the diversity of APIs proposed by different browser kernel-extension solutions will, in contrast, limit the large-scale promotion of Web AR applications. Fortunately, some standardization efforts have already begun (WebXR Editor's Draft, W3C, March 7, 2018).

[3]Average forward pass time for bvlc_googlenet with cuDNN using batch_size:128 on a K40c is 562.841 ms (Source: https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet).

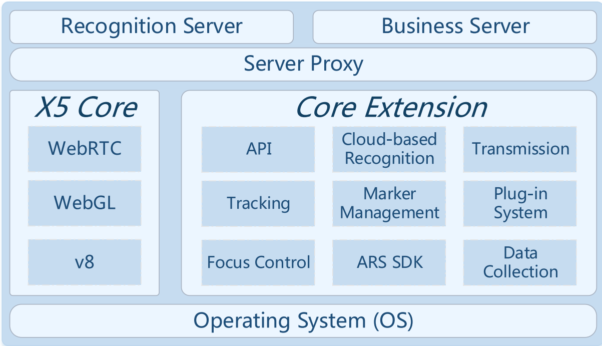


Fig. 8. ***TBS AR system architecture. By extending Web browsers to support AR, Web AR applications can provide users with near-native application performance.***

B. Computation Outsourcing

Although the browser-kernel extension method achieves a skip-type performance improvement compared with another self-contained Web AR implementation solution (i.e., pure JavaScript library/plug-in), it is still challenging for the perception of complex environments by mobile devices due to their limited computational capability. Another type of Web AR implementation mechanism is outsourcing the computations. By outsourcing computationally intensive tasks to cloud servers, Web users can get a better AR experience, which benefits from the stronger computing capability of the servers. Meanwhile, it also reduces the computing capability requirement for the mobile device and, thus, the threshold of the promotion of Web AR. However, the additional communication delay and deployment cost are two important issues that deserve our attention at the same time.

Advances in network technology make it possible not only to outsource computationally intensive tasks to cloud servers but also to achieve collaborative computing for a better AR experience and savings of energy. The emerging 5G networks can achieve even a 1-Gb/s data rate as well as millisecond end-to-end delay, and moreover, the D2D technology supports short-distance communication. All these features provide opportunities to Web AR for its pervasive promotion and performance improvement as well.

Another important issue lies in the offloading strategy for the computation outsourcing method. Considering the high monetary cost of the deployment of cloud servers, a reasonable service deployment and computation offloading method is therefore necessary. Nowadays, a variety of offloading frameworks [66]–[69] and approaches (e.g., game theory, integer linear programming, and multicriteria decision theory, and reinforcement learning) are available, which can be used for the deployment of Web AR to fulfill the adaptive computing paradigm and thus optimize the resource utilization on the Internet.

A computation outsourcing mechanism provides an alternative service provisioning paradigm for Web AR. In this section, two kinds of computation outsourcing-based Web AR implementation methods will be discussed: back end and collaborative.

1) Back-End Solutions: Compared with mobile devices (e.g., smartphone or AR glass), servers in the remote/edge cloud always have more powerful computing, rendering, as well as storage capabilities, so complex tasks can be processed more quickly and efficiently. In accordance with different deployment methods of servers, the back-end solutions can be classified as cloud computing-based and edge computing-based solutions.

a) Mobile cloud computing-based solutions: This type of solution offloads computing or rendering tasks to the remote cloud servers for process acceleration. It therefore not only alleviates the computational pressure on the mobile device but also improves the performance of the Web AR applications. Many Mobile AR applications [71]–[75] have benefited a lot from this computation outsourcing paradigm. However, there are still some issues that cannot be ignored.

1) Bandwidth Challenge: The continuous image/video transmission occupies a large part of the network bandwidth, which has a bad impact on core networks.
2) Latency Challenge: An additional communication delay is added due to the data transmission, and an unstable wireless environment also aggravates the performance of Web AR applications.

The upcoming 5G networks will provide higher bandwidth and lower network delay, and therefore, they will efficiently optimize the performance of Web AR applications in the case of computation outsourcing. Another important issue lies in the processing and cost pressures caused by the high concurrence in the case of centralized service provisioning. An example is Kurento [76], one of the typical WebRTC media server implementations, which can be used for Web AR. Even just the encoding and decoding (640×480 pixels) processes in the system will occupy about 20% of the CPU,[4] not to mention the cost pressures incurred by the concurrence requirement.

b) Promising solutions based on multiaccess edge computing: The MEC paradigm in 5G networks provides an alternative Web AR service provisioning mechanism considering the bandwidth and latency challenges faced by the aforementioned centralized solution. On the one hand, real-time pose estimation in Web AR application, either through positioning techniques or through the camera view, or both, imposes strict requirements for computing platforms. On the other hand, the "virtual contents" are always relevant to user surroundings (i.e., "reality"). Hosting the AR service on an MEC platform instead of in the cloud is advantageous as follows.

1) From Computing Aspect: Pose tracking and even rendering can be performed on an MEC platform for quality-of-service improvement.

[4]Intel Xeon CPU E5-2682 V4 @ 2.50 GHz.

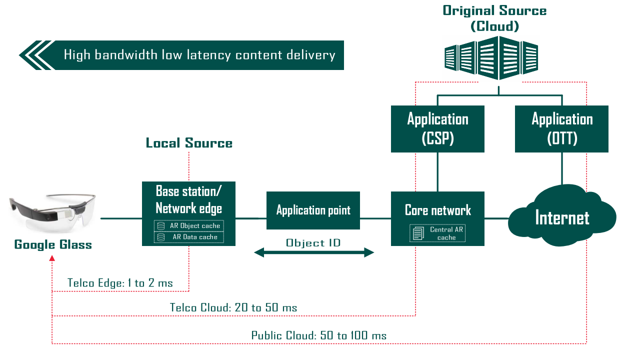


Fig. 9. ***Network topology significantly affects the content delivery latency and, thus, the user experience. 5G latency can be broken down into three different network topologies, and their associated latencies in the diagram are only from an assumption of a round-trip ping scenario (Source: ABI Research [70]).***

2) From Caching Aspect: Highly localized supplementary information on edge nodes improves the overall storage efficiency of the system.

By migrating Web AR services from the remote cloud to the network edge, where it is closer to the users, this type of solution not only reduces the communication delay but also alleviates the bandwidth usage of core networks at the same time. Additionally, this MEC-based solution also has the advantages of collecting metrics, anonymized metadata, and so on, which provide a basis for further user experience optimization. Nowadays, some efforts have already started to explore the potential of edge computing for Mobile AR applications and have achieved positive results, for example, remote live support [77]. For simplicity, here, we categorize both Cloudlets [78] and newly emerged fog computing paradigm [79]–[81] as a specific type of MEC method in this paper. The ETSI has already sketched an MEC-based AR service provisioning scenario [15] in 2015, and a general distributed network topology [70] in 5G networks is shown in Fig. 9. In the meantime, there are already some efforts on this promising computing paradigm [82]–[88]. Because of the native support of MEC technology in 5G networks, the development of Web AR services will become easy and convenient.

2) Promising Collaborative Solutions: The aforementioned self-contained Web AR implementation method faces limited computing capability, which limits its broad application. Moreover, the diversity of computing platforms also hinders the pervasive promotion of Web AR due to the lack of an adaptive computing resource scheduling mechanism. AR is a computation- and data-intensive application, as mentioned earlier, although mobile cloud computing- and MEC-based implementation methods provide more powerful computing capability, and thus, a Web AR performance improvement, the bandwidth usage, communication delay, and deployment cost all deserve our attention in the case of high concurrence requirement. On the other hand, advances in mobile devices also make it possible to perform computational tasks locally, and although current mobile devices cannot afford overly complex computational works, it still encourages us to carry out further research to explore the potential of collaborative distributed computing for Web AR. To take full advantage of distributed and diverse computational and storage resources, an adaptive and scalable collaborative computing and communication paradigm is therefore necessary. By distributing the computational pressure from cloud servers to mobile devices and network edges while still satisfying the performance requirements and user experience of a Web AR application, it can effectively gather distributed resources and then achieve cost saving and further performance improvement.

Osmotic computing [89]–[91] is a novel computing paradigm that aims to facilitate highly distributed and federated computing environments. It enables the automatic deployment of microservices over an interconnected cloud datacenter and an edge datacenter. In the meantime, the proposed reverse offloading method, i.e., the movement of functionalities from the cloud to the network edges, not only helps latency-sensitive applications but also minimizes the amount of data that must be transferred over the network. This adaptive and scalable paradigm will, therefore, be a promising direction for distributed collaborative Web AR implementations.

a) Terminal + Cloud collaborative solution: The reverse offloading method in the osmotic computing paradigm encourages us to offload part of the computational tasks of Web AR from the cloud to user devices to alleviate both computing and deployment cost pressure on the central site. We conducted a real Web AR advertising campaign for China Mobile by WeChat (December 5–14, 2017), which is also the first time we promoted Web AR on a large scale. It achieves 3 550 162 page views and 2 080 396 unique visitors in only 10 days. In this project, the Web AR service provisioning mechanism was designed in two parts. All the visual operations are executed on the mobile device locally based on the ORB algorithm in JSFeat. The remote cloud servers are responsible for the database queries or other logical tasks. The collaboration between the computing capability of the terminal (mobile device) and the storage capability of the cloud server greatly reduces the overall deployment cost of the Web AR application. However, there is still a lot of room for performance optimization, since the choice of JSFeat aims at reducing the threshold of the computing capability requirement so as to reach more Web users; it compromises the performance of the application. Yang *et al.* [92] discussed the optimization of computation partitioning for a data stream application between the terminal and the cloud. The results show that a reasonable computation offloading method provides great benefits to the performance of the application. Moreover, in accordance with the differing computational capabilities of mobile devices,

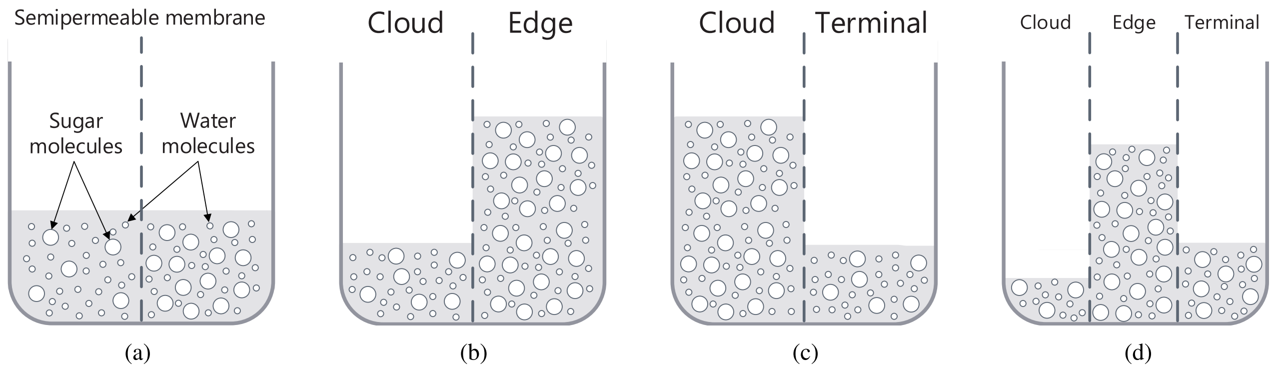


Fig. 10. ***Basic concepts of osmotic computing as well as two osmotic scenarios. Terminal + Cloud (Osmotic scenario I) and Terminal + Edge + Cloud (Osmotic scenario II) collaborative Web AR implementation approaches. (a) Osmotic concept. (b) Osmotic computing. (c) Osmotic scenario I. (d) Osmotic scenario II.***

an adaptive and scalable computation offloading method can schedule distributed resources on the Internet more flexibly and intelligently, which is important for this type of collaborative Web AR implementation solution, since it can perform personalized computation partitioning, as shown in Fig. 11(a), and hence maximize the individual user experience for Web AR applications.

b) Terminal + Edge + Cloud collaborative solution: Another promising, more efficient, but complex, method is to combine the computing and storage capabilities of the mobile devices, network edges, and remote cloud servers to explore more adaptive and scalable collaborative Web AR implementation methods. In general, the introduction of mobile devices alleviates the computational pressure on the edge and cloud servers; network edges provide a temporary place for Web AR application migration, not only supplementing the computing capability of the mobile device but also shortening the data transmission; a remote cloud server has stronger computing capability and adequate storage space and is generally responsible for the database queries, historical big data-based model training, and other tasks. In addition to the aforementioned osmotic computing paradigm which provides guidelines for this collaborative Web AR implementation solution, with the advance of the smart city, smart home, and smart devices, the user's ambient environment is becoming powerful and intelligent, and collaborative solutions help mobile devices extend their capabilities in a more flexible manner, which therefore further facilitates the promotion of Web AR. As shown in Fig. 11(b), devices communicate through the cellular or WLAN technology to share supplementary information (e.g., 3-D model, sound, and video) and collaborate to perform Web AR applications as well. Moreover, this short-distance wireless communication technology can also be used for collaborative multiuser Web AR applications, such as multiplayer online AR games, but only impose a slight performance impact on the central site. The upcoming 5G networks promise the supports of D2D technology, and apparently, the Web AR will benefit from it in more and more scenarios (e.g., museum, art gallery, and city monument). As a demonstration, we implemented a Web AR-based animal retrieval application for tourists in the zoo that leverages distributed deep neural networks (DNNs) and evaluated

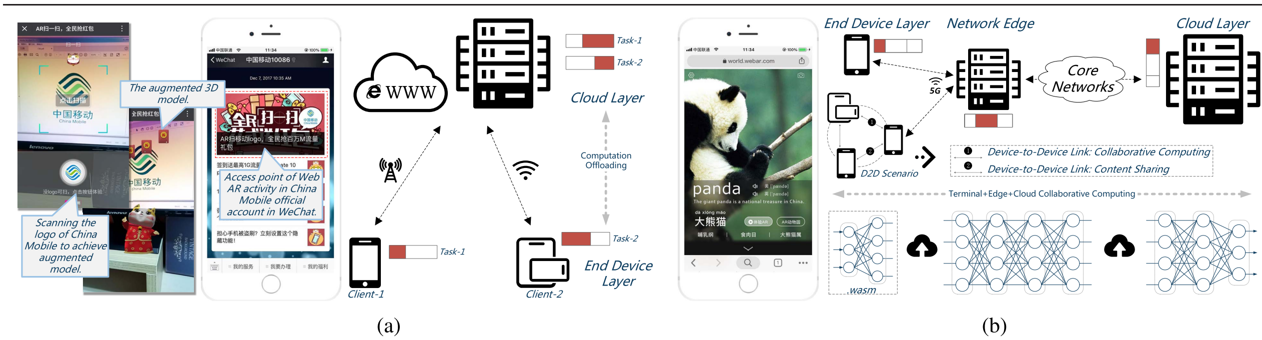


Fig. 11. ***Two promising collaborative Web AR implementation solutions and the distribution of workload. (a) Real Web AR advertising campaign for China Mobile by WeChat in current mobile networks and the illustration of the adaptive and scalable computation offloading approach. (b) Collaborative computing scenario between hierarchical platforms for Web AR and an experimental application designed based on distributed DNNs in the 5G trial networks.***

			3G / 4G				5G			
			Sensor-based Mechanism (Section II-B-1)	Vision-based Mechanism (Section II-B-2): Marker-based Method	Vision-based Mechanism (Section II-B-2): Markerless Method	Hybrid Tracking Mechanism (Section II-B-3)	Sensor-based Mechanism (Section II-B-1)	Vision-based Mechanism (Section II-B-2): Marker-based Method	Vision-based Mechanism (Section II-B-2): Markerless Method	Hybrid Tracking Mechanism (Section II-B-3)
Self-contained Method		Pure JavaScript Library/Plug-in (Section III-A-1)	+ +	+ +	+	+	+ + +	+ +	+	+
Self-contained Method		Extending the Browser Kernel (Section III-A-2)	+ + +	+ +	+	+ +	+ + +	+ + +	+ +	+ + +
Computation Outsourcing	Back-end Solutions (Section III-B-1)	Mobile Cloud Computing (MCC) -based Solutions	+ + +	+ +	+	+	+ + +	+ + +	+ +	+ +
Computation Outsourcing	Back-end Solutions (Section III-B-1)	Multi-access Edge Computing (MEC) -based Solutions	+	+ +	+	+	+ +	+ + +	+ + +	+ + +
Computation Outsourcing	Promising Collaborative Solutions (Section III-B-2)	Terminal+Cloud Collaborative Solution	+	+ +	+	+	+ +	+ + +	+ +	+ +
Computation Outsourcing	Promising Collaborative Solutions (Section III-B-2)	Terminal+Edge+ Cloud Collaborative Solution	+	+	+	+	+ +	+ +	+ + +	+ + +

Technology Readiness Levels: + = Low | ++ = Medium | +++ = High

Fig. 12. ***Technology readiness levels for Web AR system in the current 3G/4G and upcoming 5G era.***

in the 5G networks.[5] The end device and network edge servers shoulder totally 72.2% (17.6% + 54.6%) of DNN (AlexNet [93]) computations (i.e., the relieved computation pressure of the central site). Note that the specific neural network computations are encoded into the Wasm format in advance to achieve inference acceleration on the Web. In the meantime, this collaborative solution also brings the response delay improvement about 319.26% compared with the pure front-end (Chrome)-based solution. Obviously, for collaborative computation outsourcing solutions, an adaptive and scalable scheduling method will always benefit Web AR applications as the different available computing and storage resources of devices can be coordinated more intelligently.

C. Technology Readiness Levels

Based on the previous statements, we summarize and compare the technology readiness levels of Web AR implementation mechanisms and approaches, as shown in Fig. 12. Obviously, the Web AR is still in its infancy, which requires efforts from both academia and industry. Considering the challenges of computing capability, networking, battery capacity, and compatibility, the pervasive promotion and application for Web AR are still difficult in the current mobile networks as: 1) the sensor-based implementation mechanism cannot provide the seamless immersive experience to users due to the unavoidable cumulative error and 2) the browser-kernel extension solution lacks cross-platform support at present. However, the future of Web AR really deserves our expectations. The lightweight and cross-platform Web AR technologies have a wide range of applications in many areas, especially with the development of networks and the improvement of mobile device performance. The advancement of underlying technologies and the innovation of applications are of mutual promotion, and they will eventually propel the development of Web AR ecosystem.

[5]The actually deployed 5G trial network was supported by China Mobile Communications Group Beijing Co., Ltd. and Huawei Technologies Co., Ltd.

IV. OPEN RESEARCH CHALLENGES

The emergence of Web AR undoubtedly helps the promotion of Mobile AR applications on a large scale. However, there are still various obstacles waiting for the proper technologies to be available and affordable. The practical development and deployment of Web AR inspired us a lot. In this section, we detail these insights and provide some further discussions as well.

A. Computation and Rendering Efficiency

AR is a computation- and data-intensive application. However, both the computing and the rendering tasks in Web AR nowadays face an inefficient runtime environment due to the limited computational and storage abilities of mobile devices.

1) Computational Efficiency: Considering the aforementioned self-contained and collaborative computation outsourcing Web AR implementation approaches, the computational and rendering abilities of mobile devices play an important role in the improvement of the performance of Web AR applications. Here are several suggestions for the improvement of their performance.

1) WebAssembly, Web Workers, and other similar enabling Web technologies are helpful for the improvement of the performance of Web AR applications. WebAssembly can accelerate the Web AR process by transcoding high-level codes into binary format in advance, which also has the advantage of code compression at the same time. The project WebSight [94] demonstrates that WebAssembly can provide about a 10× performance improvement

over pure JaveScript. In addition, the introduction of multithread technology can also effectively improve the overall efficiency of the program by using the Web Workers technology in the Web AR application.

2) Whether it is JSFeat, JSARToolKit, or even state-of-the-art AR.js, there still are obvious performance weaknesses for natural feature tracking-based and markerless Web AR implementation methods. That is, to improve the Web AR application performance, we also need to pay attention to a more efficient computing paradigm and Web AR-related JavaScript library/plug-in.
3) Approximate computing is another way that is worth trying out. The performance of Web AR applications can be improved by reducing the complexity of the algorithms. Although some efforts [95]–[97] have already proved the feasibility of this computing paradigm, where the tolerance of imprecise operations (e.g., image recognition and motion sensing) can help the improvement of the Web AR user experience, there is still a lot of room for further investigation of approximate computing, especially in terms of Web AR.
4) Another suggestion is our aforementioned computation outsourcing Web AR implementation approach. Both back-end and collaborative solutions can provide a better user experience. An adaptive and scalable collaboration strategy will benefit the whole Web AR application provisioning framework since the computing and storage resources can be fully scheduled and utilized in an intelligent way.

2) Rendering Efficiency: Rendering efficiency is another area of concern. Virtual contents (e.g., 3-D model) generated by the computer can currently only support simple interactions with users, such as rotating and scaling operations, on the Web. Indeed, a complex 3-D model not only adds download time from the cloud/network edge but also increases the computational burden on the mobile devices. Moreover, the longer rendering time will even degrade the user experience of Web AR applications. Zhang *et al.* [98] analyzed the time and energy consumption of each part of a Mobile AR application, noting that as the complexity of the 3-D model increases, the proportion of the rendering part will also increase. For example, the 3-D model that is larger than 4 or 5 MB will result in a serious lag phenomenon in our experimental project and further degrade the Web AR application performance. It is obvious that Mobile AR, especially Web AR applications, needs more lightweight 3-D models or even a dedicated lightweight 3-D model format for Web AR. A model compression technology can only shorten the download time; the rendering operations on mobile devices still consume a large amount of CPU, memory, and battery resources. In addition, optimized rendering techniques or GPU-based rendering methods can also improve the rendering efficiency on the Web. On the other hand, by using the state-of-the-art visual attention mechanism [99], only the part of the user's attention in the field of view will be augmented, which therefore also reduces the complexity of the rendering.

B. Network Communication Efficiency

Another crucial problem for Web AR is network requirements. To achieve a higher quality of the user's experience, computationally intensive tasks are usually outsourced to cloud/network edge servers for performance improvement considering the limited computing and rendering capability of mobile devices. The aforementioned MEC paradigm can further lower the communication delay for Web AR application. However, the deployment of an edge computing system has a high monetary cost, and the infrastructures have not yet been popularized in the current 3G/4G mobile networks. A more practical way is to use currently available network technologies, such as content delivery network and data centers, to compensate for the gap between users' ever-increasing demand for Web AR applications and the computation outsourcing method under the current mobile networks. The soon-to-be available 5G networks together with their newly emerging features will bring new opportunities for the promotion of Web AR. The network slice technology provides a more reasonable network resource scheduling mechanism, which will therefore provide a better network environment for Web AR. The MEC and D2D technologies will facilitate the service provisioning on the Web in a more flexible way based on the adaptive and scalable computing and communication paradigm.

C. Energy Efficiency

AR applications require long-time cooperation of environment perception, interaction perception, and Internet connection. All these power-hungry tasks place tremendous pressure on the battery in mobile devices. However, currently, the battery is only designed for common functionalities. To reduce the adverse impact of Web AR applications on mobile devices, energy efficiency is also an important part that cannot be ignored. Multicore CPUs consume less energy than single-core CPUs due to the lower frequency and voltage, and there are already many off-the-shelf multicore CPU processors available for mobile devices. By parallelizing the tasks in a Web AR application to multicores, the energy consumption can be reduced. Moreover, the upcoming 5G networks can also help energy saving indirectly, since both the network latency and the cost of data transmission can be optimized.

D. Compatibility

Web AR is designed as a lightweight and cross-platform Mobile AR implementation to achieve the pervasive promotion of AR applications. However, the compatibility issue is also one of the most serious problems at the moment.

1) Enabling Technology Compatibility: Various browsers, including native browsers (e.g., Chrome, Firefox, and Safari) and built-in browsers in which the application is designed in a hybrid way (e.g., Facebook, Twitter, and WeChat), present a great difference in their support for and compatibility with all types of Web AR enabling technologies, such as WebAssembly, WebGL, WebRTC, and so on. This not only hinders the large-scale promotion of Web AR applications but also increases the difficulty of program development.

2) Web AR Browser Compatibility: The lack of standardization of browsers for Web AR causes another compatibility issue. Currently, all the dedicated Web AR browsers are isolated from each other; an AR application designed based on a specific Web browser cannot be accessed on other platforms. The W3C group has made some efforts [100] recently, and with the ever-growing enthusiasm for the Web AR from users, standardization also needs attention, which requires a joint effort from both academia and industry.

3) 3-D Model Format Compatibility: There are also compatibility issues between Three.js and Web 3-D models that are generated by different tools (e.g., 3DMax, MAYA, and Blender), which cause degradation of the animation effects of Web AR applications. For the purpose of being cross-platform, standardization of a Web 3-D model format is also expected.

E. Privacy and Security

Social acceptance of Web AR is easily affected by privacy and security factors. The "Stop the Cyborgs" movement has had a huge impact on Google Glass due to the privacy leakage. Whether it is a client–server (i.e., back end) or a collaborative Web AR implementation method, there are various potential invasion sources. Users' private information, such as personal identification and location information, can possibly be collected by third parties for other uses. To guarantee privacy safety, both trust mechanisms for data generation and certification mechanisms for data access, as well as a secure network environment for data transmission, are required. Acquisti *et al.* [101] discussed privacy issues of facial recognition for AR applications and also proposed several privacy guidelines including openness, individual participation, use limitation, purpose specification, and so on, as well as a recommended solution: regulate usage, not collection. Besides the standard security strategies, such as on-device and network encryption, others will need to be rethought, especially in this new context [102], [103]. For example, researchers have begun considering the specific AR operating system [104] (from underlying platform perspective), the surroundings information collection rules or retention policies [105] (from sensing perspective), the object access governing [106], [107] (from data access perspective), and the trusted renderer [108], [109] (from output perspective). In addition to these technical solutions, the privacy and security challenges for AR systems also call for social, policy, or legal approaches [110]. Web AR is more dependent on mobile networks and therefore more likely to be invaded. This poses a significant challenge for the development and deployment of Web AR applications.

F. Application Deployment

Web AR has a great potential to enrich our ways of interacting with the real world. There is a growing demand for the mobility-aware, lightweight, and cross-platform AR applications. Google Glass was a milestone product, which not only raised public interest but also played an important role in the promotion of AR, especially Mobile AR. Although most existing Web AR applications are research prototypes, the popularity of Pokémon GO has demonstrated the attraction and the potentially wide deployment of Web AR applications. We believe that with more open-source software and more development platforms and educational programs for Web AR made publicly available, more Web AR prototype systems and applications will emerge. Similarly, Web AR also needs killer applications to help developers and users explore its potential value.

V. CONCLUSION

We have presented a survey of Web AR in three focused subject areas. First, we reviewed the principle of Mobile AR and three typical implementation approaches. Second, we discussed the challenges and enabling technologies for when AR meets the Web and described different Web AR implementation approaches. Finally, we summarized the ongoing challenges and future research directions of Web AR. Although Web AR is still in its infancy, the state-of-the-art research and development results and the different Web AR implementation approaches discussed in this paper will provide guidelines and a reference entry for researchers and developers to apply Web AR technology in their Web-based mobile applications to provide a pervasive AR experience to the users. Recently, the Web-based AR implementation method has also received focused attention from the W3C group, and the Web XR Editor's draft was released in March 2018.

The upcoming 5G networks provide an efficient and powerful platform for the pervasive promotion of Web AR. The higher data rate (0.1∼1 Gb/s) and lower network delay (1∼10 ms) satisfy quite well the real-time interaction requirements of Web AR. The MEC paradigm reveals a new trend of computing paradigms, that is, a reverse offloading mechanism (e.g., osmotic computing). With the deployment of edge servers, an adaptive and scalable communication and collaboration between the cloud and network edges, as well as between edge servers and mobile devices, will provide ubiquitous capability to leverage the distributed and heterogeneous computing and storage resources and fulfill the high demands of Web AR with respect to performance improvement and energy saving. Moreover, the D2D technology provides an efficient collaborative communication solution between the devices, and network slicing can further optimize the data transmission for Web AR. We conjecture that continued advances in all these computing and networking technologies mentioned earlier will further fuel the research, development, and deployment of the Web AR-enabled service provisioning at a higher level.

The different Web AR implementation approaches we discussed in this paper provide opportunities to apply Web AR applications in practice. ■

REFERENCES

[1] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," *Proc. SPIE*, vol. 2351, pp. 282–293, Dec. 1995.
[2] R. T. Azuma, "A survey of augmented reality," *Presence, Teleoper. Virtual Environ.*, vol. 6, no. 4, pp. 355–385, 1997.
[3] M. Billinghurst, A. Clark, and G. Lee, "A survey of augmented reality," *Found. Trends Hum.-Comput. Interact.*, vol. 8, nos. 2–3, pp. 272–273, 2015.
[4] Y.-G. Kim and W.-J. Kim, "Implementation of augmented reality system for smartphone advertisements," *Int. J. Multimedia Ubiquitous Eng.*, vol. 9, no. 2, pp. 385–392, 2014.
[5] Z. Shi, H. Wang, W. Wei, X. Zheng, M. Zhao, and J. Zhao, "A novel individual location recommendation system based on mobile augmented reality," in *Proc. IEEE Int. Conf. Identificat., Inf., Knowl. Internet Things (IIKI)*, Oct. 2015, pp. 215–218.
[6] B. Perry, "Gamifying French language learning: A case study examining a quest-based, augmented reality mobile learning-tool," *Procedia-Social Behav. Sci.*, vol. 174, pp. 2308–2315, Feb. 2015.
[7] N. Gavish *et al.*, "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks," *Interact. Learn. Environ.*, vol. 23, no. 6, pp. 778–798, 2015.
[8] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.
[9] (Sep. 2016). *Pokémon GO Report*. Accessed: Mar. 5, 2018. [Online]. Available: https://pokemongolive.com/en/post/headsup
[10] (Jan. 2017). *Augmented/Virtual Reality Report 2017*. Accessed: Feb. 17, 2018. [Online]. Available: https://www.digi-capital.com/reports
[11] (Mar. 2017). *Cisco Visual Networking Index (VNI): Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper*. Accessed: Dec. 19, 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html
[12] A. Charland and B. Leroux, "Mobile application development: Web vs. native," *Commun. ACM*, vol. 54, no. 5, pp. 49–53, May 2011.
[13] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
[14] A. Reznik *et al.*, "Developing software for multi-access edge computing," ETSI, Sophia Antipolis, France, White Paper 20, 2017.
[15] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper 11, 2015, p. 1–16.
[16] P. Belimpasakis, P. Selonen, and Y. You, "Bringing user-generated content from Internet services to mobile augmented reality clients," in *Proc. IEEE Cloud-Mobile Converg. Virtual Reality Workshop (CMCVR)*, Mar. 2010, pp. 14–17.
[17] Z. Sun, A. Purohit, S. Pan, F. Mokaya, R. Bose, and P. Zhang, "Polaris: Getting accurate indoor orientations for mobile devices using ubiquitous visual patterns on ceilings," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, p. 14.
[18] R. Wang, F. Zhao, H. Luo, B. Lu, and T. Lu, "Fusion of Wi-Fi and Bluetooth for indoor localization," in *Proc. 1st Int. Workshop Mobile Location-Based Service*, 2011, pp. 63–66.
[19] D. Schmalstieg *et al.*, "Managing complex augmented reality models," *IEEE Comput. Graph. Appl.*, vol. 28, no. 4, pp. 48–57, Jun. 2007.
[20] A. D. Cheok *et al.*, "Human pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area," in *Proc. Int. Conf. Mobile Hum.-Comput. Interact.* Springer, 2003, pp. 209–223.
[21] K. Greene, M. Kahari, and D. Murphy, "Hyperlinking reality via phones," *Technol. Rev.*, pp. 1–4, Nov. 2006.
[22] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.
[23] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. 2nd IEEE ACM Int. Workshop Augmented Reality (IWAR)*, Oct. 1999, pp. 85–94.
[24] S. Deffeyes, "Mobile augmented reality in the data center," *IBM J. Res. Develop.*, vol. 55, no. 5, pp. 1–5, 2011.
[25] M. Mohring, C. Lessig, and O. Bimber, "Video see-through AR on consumer cell-phones," in *Proc. 3rd IEEE/ACM Int. Symp. Mixed Augmented Reality*, Nov. 2004, pp. 252–253.
[26] T. Ball and S. K. Rajamani, "The SLAM toolkit," in *Proc. Int. Conf. Comput. Aided Verification*. Springer, 2001, pp. 260–264.
[27] R. Paucher and M. Turk, "Location-based augmented reality on mobile phones," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2010, pp. 9–16.
[28] Z. Hu and K. Uchimura, "Fusion of vision, GPS and 3D gyro data in solving camera registration problem for direct visual navigation," *Int. J. ITS*, vol. 4, no. 1, pp. 3–12, 2006.
[29] P. Honkamaa, S. Siltanen, J. Jáppinen, C. Woodward, and O. Korkalo, "Interactive outdoor mobile augmentation using markerless tracking and GPS," in *Proc. Virtual Reality Int. Conf. (VRIC)*, Laval, France, 2007, pp. 285–288.
[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.
[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[32] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan, and B. Aboba, "Webrtc 1.0: Real-time communication between browsers," Working Draft W3C, 2012, vol. 91.
[33] (2017). WebAssembly Community Group. *WebAssembly Specification*. Accessed: Mar. 13, 2018. [Online]. Available: https://webassembly.github.io/spec/
[34] A. Haas *et al.*, "Bringing the web up to speed with WebAssembly," in *Proc. 38th ACM SIGPLAN Conf. Program. Language Design Implement.*, 2017, pp. 185–200.
[35] S. Taheri, A. Veidenbaum, A. Nicolau, and M. R. Haghighat, "OpenCV.js: Computer vision processing for the Web," Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2017.
[36] S. Taheri, A. Vedienbaum, A. Nicolau, N. Hu, and M. R. Haghighat, "OpenCV.js: Computer Vision processing for the open Web platform," in *Proc. 9th ACM Multimedia Syst. Conf.*, 2018, pp. 478–483.
[37] A. Møller, "Technical perspective: WebAssembly: A quiet revolution of the Web," *Commun. ACM*, vol. 61, no. 12, p. 106, 2018.
[38] I. Hickson, "Web workers: Editor's draft," Tech. Rep., 2015.
[39] C. Marrin, "WEBGL specification," Khronos WebGL Working Group, 2011.
[40] B. Danchilla, "Three.js framework," in *Beginning WebGL for HTML5*. Springer, 2012, pp. 173–203.
[41] J. Etienne. (2017). *AR.js Project Homepage*. Accessed: Feb. 13, 2018. [Online]. Available: https://github.com/jeromeetienne/AR.js
[42] *Awe.js Project Homepage*. Accessed: Mar. 19, 2018. [Online]. Available: https://awe.media
[43] J. Mellado. *Js-ArUco—A JavaScript Library for Augmented Reality Applications—Google Project Hosting*. Accessed: Mar. 7, 2018. [Online]. Available: http://code.google.com/p/js-aruco
[44] I. Heikkinen. (2011). *JSARToolKit—JavaScript Port of FLARTOOLKit*. Accessed: Feb. 12, 2018. [Online]. Available: https://github.com/kig/JSARToolKit
[45] I. P. H. Kato, M. Billinghurst, and I. Poupyrev, "Artoolkit user manual, version 2.33," Feb. 2000.
[46] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[47] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 404–417.
[48] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2937–2944.
[49] A. Karpathy. (2014). *ConvNetJS: Deep Learning in Your Browser*. Accessed: Mar. 23, 2018. [Online]. Available: http://cs.stanford.edu/people/karpathy/convnetjs
[50] C. Körner. *CaffeJS: Run Caffe Models in the Browser Using ConvNetJS*. Accessed: Mar. 23, 2018. [Online]. Available: https://github.com/chaosmail/caffejs
[51] M. Hidaka, Y. Kikura, Y. Ushiku, and T. Harada, "WebDNN: Fastest DNN execution framework on Web browser," in *Proc. ACM Conf. Multimedia*, 2017, pp. 1213–1216.
[52] S. Daniel, T. Nikhil, and N. Charles. (2017). *Deeplearn.js: A Hardware-Accelerated Machine Intelligence Library for the Web*. Accessed: Mar. 24, 2018. [Online]. Available: https://deeplearnjs.org/
[53] (2018). *TensorFlow.js: A WebGL Accelerated, Browser Based JavaScript Library for Training and Deploying ML Models*. Accessed: Apr. 24, 2018. [Online]. Available: https://js.tensorflow.org/
[54] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[55] W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2285–2294.
[56] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[58] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 525–542.

[59] R. Kooper and B. MacIntyre, "Browsing the real-world wide Web: Maintaining awareness of virtual information in an AR information space," *Int. J. Hum.-Comput. Interact.*, vol. 16, no. 3, pp. 425–446, 2003.

[60] S. Perry, "Wikitude: Android app with augmented reality: Mind blowing," *Digit.-Lifestyles Info*, vol. 23, no. 10, 2008.

[61] B. MacIntyre, A. Hill, H. Rouzati, M. Gandy, and B. Davidson, "The Argon AR Web browser and standards-based AR application environment," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2011, pp. 65–74.

[62] *Mozilla WebXR Project Homepage*. Accessed: Feb. 14, 2018. [Online]. Available: https://deeplearnjs.org/

[63] *Google AR Project Homepage*. Accessed: Nov. 16, 2017. [Online]. Available: https://github.com/google-ar

[64] *Baidu DuMix AR Project Homepage*. Accessed: Mar. 23, 2018. [Online]. Available: https://cloud.baidu.com/product/ar.html

[65] *Tencent TBS AR Project Homepage*. Accessed: Dec. 7, 2017. [Online]. Available: https://x5.tencent.com/tbs/guide/tbsar.html

[66] H.-H. Chu, H. Song, C. Wong, S. Kurakake, and M. Katagiri, "Roam, a seamless application framework," *J. Syst. Softw.*, vol. 69, no. 3, pp. 209–226, 2004.

[67] E. Cuervo *et al.*, "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 49–62.

[68] K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 73–84.

[69] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Conf. Comput. Syst.*, 2011, pp. 301–314.

[70] (Jan. 2017). *Augmented and Virtual Reality: The First Wave of 5G Killer Apps*. [Online]. Available: https://www.qualcomm.com/documents/augmented-and-virtual-reality-first-wave-5g-killer-apps

[71] C.-F. Chang and S.-H. Ger, "Enhancing 3D graphics on mobile devices by image-based rendering," in *Proc. Pacific-Rim Conf. Multimedia*. Springer, 2002, pp. 1105–1111.

[72] G. Fritz, C. Seifert, and L. Paletta, "A mobile vision system for urban detection with informative local descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. Syst. (ICVS)*, Jan. 2006, p. 30.

[73] W. Pasman and C. Woodward, "Implementation of an augmented reality system on a PDA," in *Proc. 2nd IEEE ACM Int. Symp. Mixed Augmented Reality*, Oct. 2003, pp. 276–277.

[74] F. Lamberti, C. Zunino, A. Sanna, A. Fiume, and M. Maniezzo, "An accelerated remote graphics architecture for PDAS," in *Proc. 8th Int. Conf. 3D Web Technol.*, 2003, pp. 55–61.

[75] F. Lamberti and A. Sanna, "A streaming-based solution for remote visualization of 3D graphics on mobile devices," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 2, pp. 247–260, Mar. 2007.

[76] L. L. Fernández, M. P. Díaz, R. B. Mejías, F. J. López, and J. A. Santos, "Kurento: A media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2013, pp. 1–6

[77] M. Schneider, J. Rambach, and D. Stricker, "Augmented reality based on edge computing using the example of remote live support," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 1277–1282.

[78] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proc. 3rd ACM Workshop Mobile Cloud Comput. Services*, 2012, pp. 29–36.

[79] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.

[80] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[81] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[82] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[83] A. Glikson, S. Nastic, and S. Dustdar, "Deviceless edge computing: Extending serverless computing to the edge of the network," in *Proc. 10th ACM Int. Syst. Storage Conf.*, 2017, p. 28.

[84] S. Nastic *et al.*, "A serverless real-time data analytics platform for edge computing," *IEEE Internet Comput.*, vol. 21, no. 4, pp. 64–71, 2017.

[85] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, Nov. 2014.

[86] M. Erol-Kantarci and S. Sukhmani, "Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G," in *Ad Hoc Networks*. Springer, 2018, pp. 169–177.

[87] X. Qiao, P. Ren, S. Dustdar, and J. Chen, "A new era for Web ar with mobile edge computing," *IEEE Internet Comput.*, vol. 22, no. 4, pp. 46–55, Jul./Aug. 2018.

[88] P. Ren, X. Qiao, J. Chen, and S. Dustdar, "Mobile edge computing—A booster for the practical provisioning approach of Web-based augmented reality," in *Proc. 3rd ACM/IEEE Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 349–350.

[89] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: A new paradigm for edge/cloud integration," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 76–83, Nov./Dec. 2016.

[90] P. Hoenisch, D. Schuller, S. Schulte, C. Hochreiner, and S. Dustdar, "Optimization of complex elastic processes," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 700–713, Sep. 2016.

[91] D. Moldovan, G. Copil, and S. Dustdar, "Elastic systems: Towards cyber-physical ecosystems of people, processes, and things," *Comput. Standards Interfaces*, vol. 57, pp. 76–82, Mar. 2018.

[92] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23–32, Mar. 2013.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[94] *Web-Sight Project Homepage*. Accessed: Mar. 28, 2018. [Online]. Available: https://github.com/Web-Sight/WebSight

[95] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. 18th IEEE Eur. Test Symp. (ETS)*, May 2013, pp. 1–6.

[96] Q. Xu, M. Todd, and S. K. Nam, "Approximate computing: A survey," *IEEE Design Test*, vol. 33, no. 1, pp. 8–22, Feb. 2016.

[97] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62-1–62-33, Mar. 2016.

[98] W. Zhang, B. Han, and P. Hui, "On the networking challenges of mobile augmented reality," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, 2017, pp. 24–29.

[99] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[100] (Mar. 2018). *WebXR Device API Specification*. Accessed: Mar. 21, 2018. [Online]. Available: https://immersive-web.github.io/webxr/

[101] A. Acquisti, R. Gross, and F. Stutzman, "Faces of facebook: Privacy in the age of augmented reality," Tech. Rep., 2011.

[102] F. Roesner, T. Kohno, and D. Molnar, "Security and privacy for augmented reality systems," *Commun. ACM*, vol. 57, no. 4, pp. 88–96, 2014.

[103] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner, "Towards security and privacy for multi-user augmented reality: Foundations with end users," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 392–408.

[104] L. D'Antoni *et al.*, "Operating system support for augmented reality applications," in *Proc. HotOS*, vol. 13, May 2013, p. 21.

[105] E. Zarepour, M. Hosseini, S. S. Kanhere, and A. Sowmya, "A context-based privacy preserving framework for wearable visual lifeloggers," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–4.

[106] M. T. I. Ziad, A. Alanwar, M. Alzantot, and M. Srivastava, "Cryptoimg: Privacy preserving processing over encrypted images," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2016, pp. 570–575.

[107] Y. A. Sekhavat, "Privacy preserving cloth try-on using mobile augmented reality," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1041–1049, May 2017.

[108] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner, "Securing augmented reality output," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 320–337.

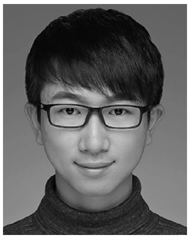
[109] S. Ahn, M. Gorlatova, P. Naghizadeh, M. Chiang, and P. Mittal, "Adaptive fog-based output security for augmented reality," in *Proc. Morning Workshop Virtual Reality Augmented Reality Netw.*, 2018, pp. 1–6.

[110] F. Roesner, T. Denning, B. C. Newell, T. Kohno, and R. Calo, "Augmented reality: Hard problems of law and policy," in *Proc. Int. Joint Conf. Pervasive Ubiquitous Comput., Adjunct Publication*, 2014, pp. 1283–1288.

ABOUT THE AUTHORS

Xiuquan Qiao is currently a Full Professor with the Beijing University of Posts and Telecommunications, Beijing, China, where he is also the Deputy Director of the Key Laboratory of Networking and Switching Technology, Network Service Foundation Research Center of State. He has authored or co-authored over 60 technical papers in international journals and at conferences, including the *IEEE Communications Magazine*, *Computer Networks*, *IEEE Internet Computing*, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the *ACM SIGCOMM Computer Communication Review*. His current research interests include the future Internet, services computing, computer vision, augmented reality, virtual reality, and 5G networks.

Dr. Qiao was a recipient of the Beijing Nova Program in 2008 and the First Prize of the 13th Beijing Youth Outstanding Science and Technology Paper Award in 2016.



Pei Ren is currently working toward the Ph.D. degree at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China.

His current research interests include the future Internet architecture, services computing, computer vision, machine learning, augmented reality, edge computing, and 5G networks.

Schahram Dustdar (Fellow, IEEE) was an Honorary Professor of Information Systems at the Department of Computing Science, University of Groningen, Groningen, The Netherlands, from 2004 to 2010. From 2016 to 2017, he was a Visiting Professor at the University of Sevilla, Sevilla, Spain. In 2017, he was a Visiting Professor at the University of California at Berkeley, Berkeley, CA, USA. He is currently a Professor of Computer Science with the Distributed Systems Group, Technische Universität Wien, Vienna, Austria.

Dr. Dustdar was an elected member of the Academy of Europe, where he is the Chairman of the Informatics Section. He was a recipient of the ACM Distinguished Scientist Award in 2009, the IBM Faculty Award in 2012, and the IEEE TCSVC Outstanding Leadership Award for outstanding leadership in services computing in 2018. He is the Co-Editor-in-Chief of the *ACM Transactions on Internet of Things* and the Editor-in-Chief of *Computing* (Springer). He is also an Associate Editor of the IEEE TRANSACTIONS ON SERVICES COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the *ACM Transactions on the Web*, and the *ACM Transactions on Internet Technology*. He serves on the Editorial Board of IEEE INTERNET COMPUTING and the *IEEE Computer Magazine*.

Ling Liu (Fellow, IEEE) is currently a Professor at the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, USA. She directs the research programs at the Distributed Data Intensive Systems Lab, examining various aspects of large-scale big data systems and analytics, including performance, availability, security, privacy, and trust. Her current research is sponsored primarily by the National Science Foundation and IBM. She has published over 300 international journal and conference articles.

Dr. Liu was a recipient of the IEEE Computer Society Technical Achievement Award in 2012 and the Best Paper Award from numerous top venues, including ICDCS, WWW, IEEE Cloud, IEEE ICWS, and ACM/IEEE CCGrid. She served as the general chair and the PC chair for numerous IEEE and ACM conferences in big data, distributed computing, cloud computing, data engineering, very large databases, and the World Wide Web fields. She served as the Editor-in-Chief for the IEEE TRANSACTIONS ON SERVICE COMPUTING from 2013 to 2016. She is the Editor-in-Chief of the *ACM Transactions on Internet Technology*.

Huadong Ma received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China, in 1995.

From 1999 to 2000, he held a visiting position at the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. He was a Visiting Professor at The University of Texas at Arlington, Arlington, TX, USA, in 2004. He is currently a Chang Jiang Scholar Professor with the State Key Laboratory of Networking and Switching Technology and the Executive Dean of the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing. He has authored over 200 papers and four books. His current research interests include Internet of Things, sensor networks, multimedia system, and networking.

Dr. Ma was a recipient of the National Funds for Distinguished Young Scientists in 2009. He serves as an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE INTERNET OF THINGS JOURNAL, and the *ACM Transactions on Internet Technology* and the Chair for the Association for Computing Machinery Special Interest Group on Mobility of Systems, Users, Data, and Computing, China.

Junliang Chen received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1955, and the Ph.D. degree in electrical engineering from the Moscow Institute of Radio Engineering, Moscow, Russia, in 1961.

He has been with the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, since 1955, where he is currently the Chairman and a Professor with the Research Institute of Networking and Switching Technology. His current research interests include communication networks and next-generation service creation technology.

Dr. Chen was elected as a member of the Chinese Academy of Sciences in 1991 and a member of the Chinese Academy of Engineering in 1994 for his contributions to fault diagnosis in stored program control exchange. He received the First, Second, and Third prizes of the National Scientific and Technological Progress Award in 1988, 2004, and 1999, respectively.