DEPARTMENT: Internet of Things, People, and Processes

A New Era for Web AR with Mobile Edge Computing

Xiuquan Qiao
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications

Pei Ren
Beijing University of Posts and Telecommunications

Schahram Dustdar
TU Wien

Junliang Chen
Beijing University of Posts and Telecommunications

Dedicated device-based and app-based augmented reality (AR) solutions have inherent limitations regarding cross-platform, pervasive AR application provisioning. *Web-based AR* (web AR), a promising lightweight and cross-platform approach to AR, is gaining increasing attention owing to its extensive application areas. However, for computationally intensive AR applications, the weak computational efficiency of current web browsers seriously hampers applications of web AR on a large scale. The "browser + cloud" approach suffers from the high-latency dilemma. Now, with the emerging 5G networks, *mobile edge computing* (MEC) promises to greatly reduce network latency (even to 1 ms) by deploying applications at the network edge closer to users, which provides an opportunity for performance improvement of web AR. In this article, the authors envision that the application of MEC has the potential to bring web AR into a new era. Specifically, an MEC-oriented web AR solution is first proposed, followed by the design and deployment details. The authors also discuss future directions aimed at using MEC to tackle the performance issues of web AR in 3G, 4G, and 5G networks.

Mobile augmented reality (MAR)[1] nowadays greatly attracts both academic and industrial attention, owing to its extensive application areas and the popularity of smartphones and wearable devices. Recently, many MAR use cases have already emerged in the fields of art, education, entertainment, and so on. These use cases typically are divided into two types: wearable-device-based and app-based MAR solutions. However, both approaches suffer from some inherent limitations regarding cross-platform, portability, and cost aspects:

1089-7801/18/$33.00 USD

- Wearable-device-based MAR solutions are costly and inconvenient to carry.
- App-based MAR solutions require downloading and installing the specific app in advance and lack cross-platform support (i.e., an AR activity of one app cannot be used in other apps directly).

Therefore, it's necessary to explore a lightweight, low-cost, and cross-platform pervasive MAR solution for mobile users. Owing to web-based AR's cross-platform and no-installation advantages, it is attracting more and more attention and is opening up a new research field of MAR.

Currently, the proposed web AR implementations can be classified primarily as pure front-end solutions using JavaScript at a mobile Web browser, browser-kernel-based extension solutions, or "browser + cloud" solutions. However, all these web AR solutions are also struggling with some problems:

- *Pure front-end solutions*. MAR applications often rely on computationally intensive computer vision algorithms. However, JavaScript cannot provide efficient computational capability for complex matrix computations. Therefore, pure front-end solutions can only use several typical algorithms to recognize some simple markers, and have poor recognition and tracking ability for nature images and real material objects.
- *Browser-kernel-based extension solutions*. To more fully utilize the computational resources of smartphones than the pure front-end solutions, some web browser providers try to extend the browser kernel to enable web AR functions, even though this approach still faces the challenges of cross-platform requirements owing to the lack of web AR standardization between different web browsers. In addition, if the application scenario needs more computing capacity (for example, to recognize more pictures), the computing capacity of end devices is still limited.
- *Browser + cloud solutions*. Considering that MAR applications often involve complex matrix computations with extreme latency requirements, the local computing operations are often limited by hardware capabilities (e.g., CPU, GPU, memory, and battery). To address this issue, cloud computing is currently used to extend the computing ability of end devices. However, this approach also brings about significant communication latency, which often incurs poor quality of experience (QoE), such as in real-time object-tracking scenarios.

On the basis of the above analysis, we need to explore a practical web AR implementation solution to resolve both the computational inefficiency of the pure front-end and browser-kernel-based solutions and the high-latency of browser + cloud solutions. Fortunately, with the emergence of 5G networks, the new *mobile edge computing* (MEC) paradigm[2] aims to provide cloud-computing capabilities at the edge of networks close to mobile users. Thus, MEC can efficiently reduce the communication latency and strengthen the end device's computing capability. We argue that MEC will provide an opportunity for the performance improvement of web AR and open up a new era for web AR applications.

In this article, we first give an overview of the state of the art of web-based MAR solutions and discuss the problems to be addressed. Then, a practical web AR service-provisioning framework leveraging the MEC paradigm is presented, which achieves encouraging performance improvement. Finally, we discuss future research directions of web AR with MEC in current 3G and 4G networks and future 5G networks.

THE STATE OF THE ART

Pure Front-End Solutions

In general, pure front-end web AR solutions utilize different JavaScript libraries to support web AR applications, which can be adopted simply in users' end devices.

AR.js[3] is one of the popular marker-based MAR solutions for efficiently performing AR applications on the Web, including recognition, tracking, and 3D object rendering on any mobile browser with WebRTC and WebGL. JSARToolKit (https://github.com/kig/JSARToolKit) is an AR library written in JavaScript, which allows rendering a 3D model of a detected marker inside a camera feed by computing the distance from the camera to the physical marker in real time. CaffeJS (https://github.com/chaosmail/caffejs) aims to run neural-network functions (e.g., image recognition) by porting Caffe models to a web browser using a modified version of ConvNetJS, which saves a lot of network traffic and server resources.

Browser-Kernel-Based Extension Solutions

Another branch of web AR solutions is browser-kernel-based extension solutions, where the functions of AR applications have an extreme dependency on browsers. Currently, most browser-kernel-based web AR projects are still in closed beta stages.

Google is developing WebARonARKit and WebARonARCore, which are experimental apps for iOS and Android that let developers create AR experiences using web technologies. Mozilla has announced its mixed-reality program: Mozilla WebXR (https://github.com/mozilla/webxr-api), which aims to make it easy for web developers to create web applications that adapt to the capabilities of each platform. The Baidu mobile-browser team has also shared its advancements in web AR and web VR in a recent open class. The Baidu mobile-browser T7 kernel supports the WebVR standard and has completed web AR prototype development (https://github.com/baidu/AR). Argon,[4] an AR-supported web browser, provides a pervasive web AR application deployment platform. Moreover, Argon implements the JavaScript library argon.js, which aims to support web AR applications in any web browser.

MOTIVATION AND PROBLEM STATEMENT

The aforementioned web AR implementations all face inherent limitations. The pure front-end solutions lack computational efficiency, and the browser-kernel-based extension solutions are still in their infancy and have not yet been adopted on a large scale. The cloud-computing paradigm greatly extends the computing ability of the terminal devices and provides an opportunity for the promotion of web AR applications on users' end devices with limited hardware resources.

Benefiting from the cloud-computing paradigm, we have conducted several online real advertising promotion activities using web AR within WeChat, which have gained a great deal of attention. According to our practical project experience, nowadays most users' end devices such as smartphones and tablet PCs have satisfied the deployment requirements of web AR. Technologies that have enabled this situation include

- WebAssembly, a new portable, size- and load-time-efficient format suitable for compilation to the web;
- Web Workers, an HTML5 technique that provides a means to run a script operation in a background thread separate from the main execution thread of a web application; and
- WebRTC, a collection of communications protocols and APIs that enable real-time communication over peer-to-peer connections.

Moreover, all devices equipped with iOS 11 or later versions support WebRTC. All these enabling techniques provide a platform for web AR applications' deployment, enabling the lightweight, cross-platform and pervasive application of web AR.

Here, we detail two influential use cases as shown in Figure 1, describing the challenges we faced during design and development.

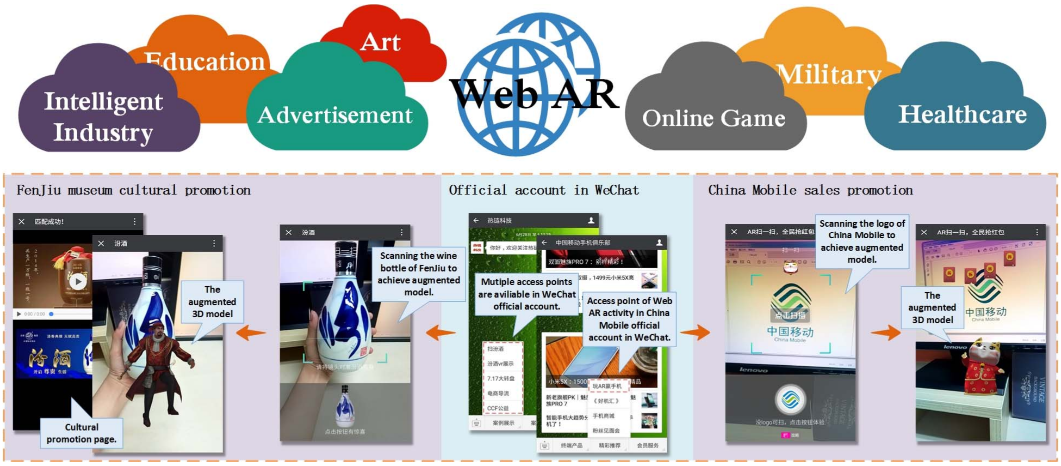

Figure 1. Two real use cases of web augmented reality (web AR) for market promotions of China Mobile and FenJiu.

- *China Mobile use case*. A web AR sales promotion for the CMCC A3s smartphone has been proposed for advertising effects. The activity URL link is embedded in China Mobile's official account in WeChat. Users only need to access the activity item in the corresponding official account, by scanning the logo of China Mobile; an augmented 3D model then is rendered on a webpage. In this application scenario, the most important and complex part is image matching. However, the available image-matching algorithms such as SURF (Speeded-Up Robust Features), SIFT (Scale-Invariant Feature Transform), and ASIFT (Affine-SIFT) are too heavy to run on web browsers. The ORB (Oriented FAST and Rotated BRIEF) matching algorithm provides a faster feature extraction method and can be used directly by JavaScript; however, it results in insufficient matching accuracy.
- *FenJiu use case*. To promote the company culture, FenJiu has adopted a web AR technique for its museum. As in the use case mentioned above, users can experience the web AR application by scanning different material objects such as wine bottles and other exhibits in the museum. However, once the image is captured by a user's end device, it needs to be compared with 10 other samples (the number of samples will increase in the future), the results of which are used to identify different exhibits and then present different cultural-promotion pages. Such intensive computing apparently cannot be performed by users' end devices locally. Benefiting from the cloud-computing paradigm, computation tasks, therefore, can be migrated to cloud servers. However, the cloud-computing approach introduces high latency due to long-distance data transmission through the wireless network, which significantly degrades the performance of the web AR effects.

These two use cases placed us in a dilemma:

- MAR applications rely on complex computer vision algorithms, while web browsers always lack sufficient computational capability.
- Although the cloud-computing paradigm extends the computational and storage capability for users' end devices, it causes significant latency in the mobile network. However, MAR applications require low latency.

WEB AR WITH MOBILE EDGE COMPUTING

By offloading computationally intensive workloads to the proximity servers at the edge of the network, the emerging edge-computing paradigm provides a promising solution to meet the ever-increasing computational demands of numerous mobile applications.[5] To compensate for the lack of computing capability and to ease the new constraints caused by cloud computing, especially concerning latency and bandwidth, we propose a web AR service-provisioning framework with MEC, on the basis of our project experiences.

Overview of the MEC Framework for Web AR

The overall framework is first proposed in three perspectives as shown in Figure 2, followed by the details of three processing components on the terminal, the edge cloud server, and the remote cloud server. We conducted the experiments on the basis of our practical projects, and the results fully demonstrate the advantages of MEC for web AR applications.

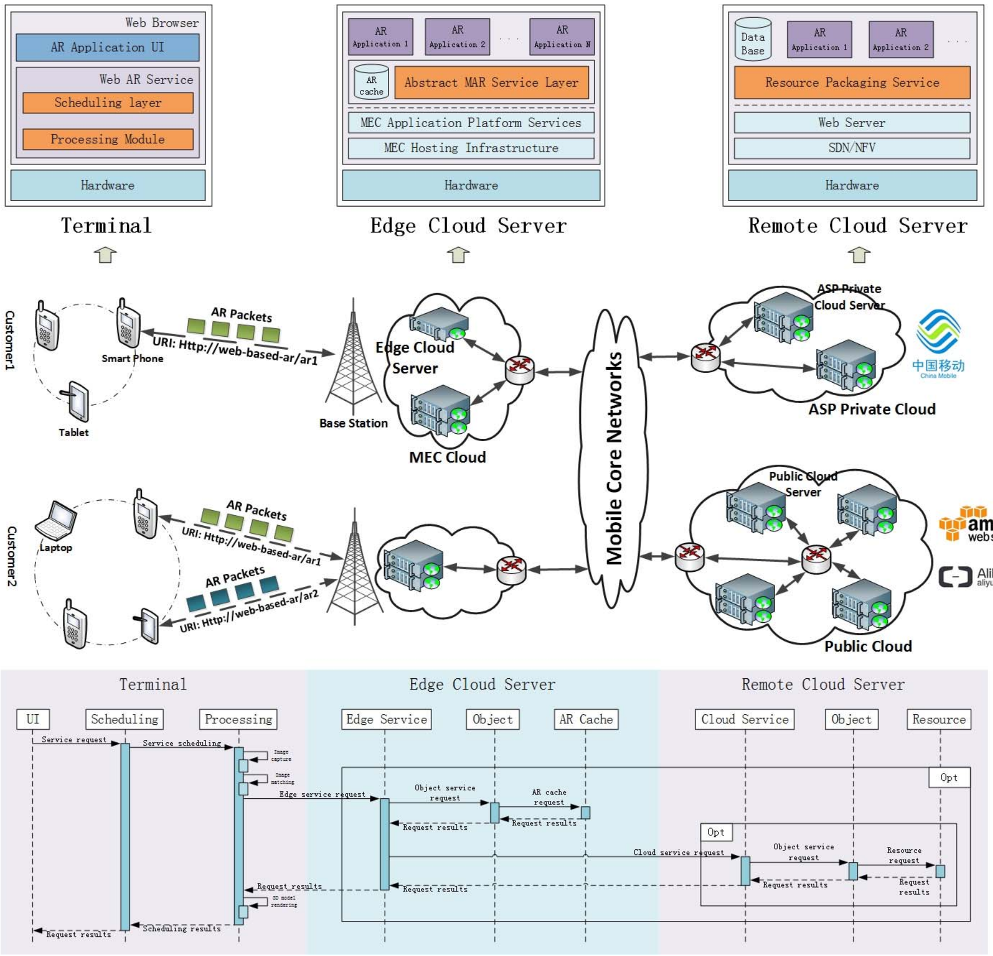

Figure 2. Overview of the framework for web AR with mobile edge computing (MEC).

The Terminal Side

The terminal-side computation platform is responsible mainly for web AR service scheduling and basic processing, which will not pose a severe computational burden, considering the weak computational capability of the web browsers in users’ end devices.

The web AR service platform consists mainly of two modules: the *scheduling layer* and *processing module*. The processing module provides the underlying support for the scheduling layer. The scheduling layer processes all the web AR application logic and collaborates with the related service parts.

The processing module in the web AR service layer is composed of three submodules: the image-capture, image-matching, and 3D-model-rendering submodules. The image-capture submodule uses a WebRTC technique to capture an image from a camera and then execute a resizing operation (set by the application service provider in advance) on it, taking into account the communication cost of image transmission. The image-matching submodule performs an image-to-image matching operation, leveraging the lightweight image-matching algorithms such as ORB mentioned above, which is already supported in JSFeat (the JavaScript Computer Vision Library). The 3D-model-rendering submodule leverages a WebGL technique to perform 3D model rendering. Users can interact with the 3D models to acquire more augmented information, which provides a more attractive and friendlier experience compared to traditional ways.

However, for some complex images, the proposed ORB matching algorithm cannot work well. Then, the specific edge cloud services will be invoked in the case where the image-matching result acquired on the terminal side is insufficient for the web AR application.

The Edge Cloud Side

The edge-cloud-side computation platform consists mainly of an *abstract MAR service layer*, which is used to process incoming edge service requests and manage the web AR application objects, including object deployment, object destruction, and other underlying service supports.

The underlying abstract MAR service layer consists of some common web AR functional modules, which aim to provide simpler and faster service to the web AR application instances in the upper layer and reduce the access cost to hardware at the same time, so as to improve the edge server's overall performance. Currently, it includes a performance-monitoring module and image-matching module.

Once the edge cloud server receives a web AR request (i.e., an image-matching request) from the terminal side, it will directly forward that request to the specific web AR application instance. The AR cache in the edge cloud server can be accessed by all web AR applications. However, in the case where the requested web AR application has not been deployed yet, the abstract MAR service layer will send the application deployment request to the remote cloud server. The current performance of the edge server, including the CPU, memory, and storage usage, is also sent to the remote cloud server for the decision making of the web AR application deployment. The remote cloud server—i.e., the application service provider (ASP)—finally determines where the specific application is to be deployed, taking into account the overall cost of deployment and transmission.

After the abstract MAR service layer in the edge server receives the requested application, it will perform deployment. Thus, it provides more efficient servicing of the follow-up web AR requests. Moreover, the processing module in the terminal performs the rendering operation on the specific 3D model according to the received matching result.

The Remote Cloud Side

The remote-cloud-side computation platform aims at providing a more generalized service provision mechanism. One of the most important components in the framework is the *resource packaging service* layer, which acts as a web AR resource manager. When the ASP faces different web AR application requirements, the resource-packaging-service layer combines specific web AR resource components into different web AR applications, which will then be deployed to appropriate edge cloud servers according to the deployment decisions. Meanwhile, different ASPs also have different web AR resources, including various 3D models, image-matching algorithms, etc. The resource-packaging-service layer is also responsible for the management of these contents.

The database is used mainly for the user information store. Moreover, images that match successfully on the edge cloud server will also be transmitted to the remote cloud server, since these types of images are currently not handled by the terminal-side image-matching algorithm. These images can be used to improve the performance of the terminal-side image-matching algorithm in the future.

Performance Evaluation

Theoretically, the introduction of the edge server will bring the following advantages:

- In scenarios that need to match a single image, the matching accuracy can be improved by providing a more optimized algorithm on the edge server side. The edge server has greater computational capability, and the image-matching time can be reduced effectively compared to the same matching algorithm on the terminal side. Although uploading images to the edge server adds extra time, the total processing efficiency still improves. Moreover, the edge server performs better since it is closer to the users. Hence, the network delay is no longer the bottleneck for web AR applications when the MEC paradigm is adopted.
- In the case where multiple images need to be compared, users' end devices apparently cannot meet the performance requirement of the web AR application owing to their limited computational capability. Migrating the service module (i.e., the web AR application) from the remote cloud server to the edge server effectively eases the issue of network delay caused by the frequent image transmissions.

To get performance comparisons, we conducted experiments on a Samsung Note 4 mobile phone, using Chrome under the use cases mentioned above. The mobile phone connected to a Wi-Fi access point on our campus (i.e., BUPT-mobile). By modifying the DNS setting, all the requests were redirected to the server in our lab, which played the role of the edge server. The MEC framework and two web AR applications were deployed in our lab and on Alibaba Cloud, respectively. A modified version of the ASIFT algorithm was employed to provide more accurate and effective image matching. Moreover, the feature points of image samples on the server were computed in advance to achieve faster matching. Four performance metrics are considered: frames per second (fps), latency, the terminal's power consumption, and the average number of matched feature points.

In experiments, the image was first resized to 250 * 250 pixels, about 8.92 Kbytes on average. The round-trip time for terminal-edge and terminal-cloud was 13.787 ms and 39.781 ms, respectively, for 64 bytes. Each ORB image-matching operation cost 394 ms on a user's end device, while only about 30 ms on the server. Hence, offloading computationally intensive tasks to the edge cloud was still a good choice. Meanwhile, edge computing performed better than the cloud paradigm regarding latency and power consumption, as shown in Figure 3, since it benefited from a shorter image transmission distance.

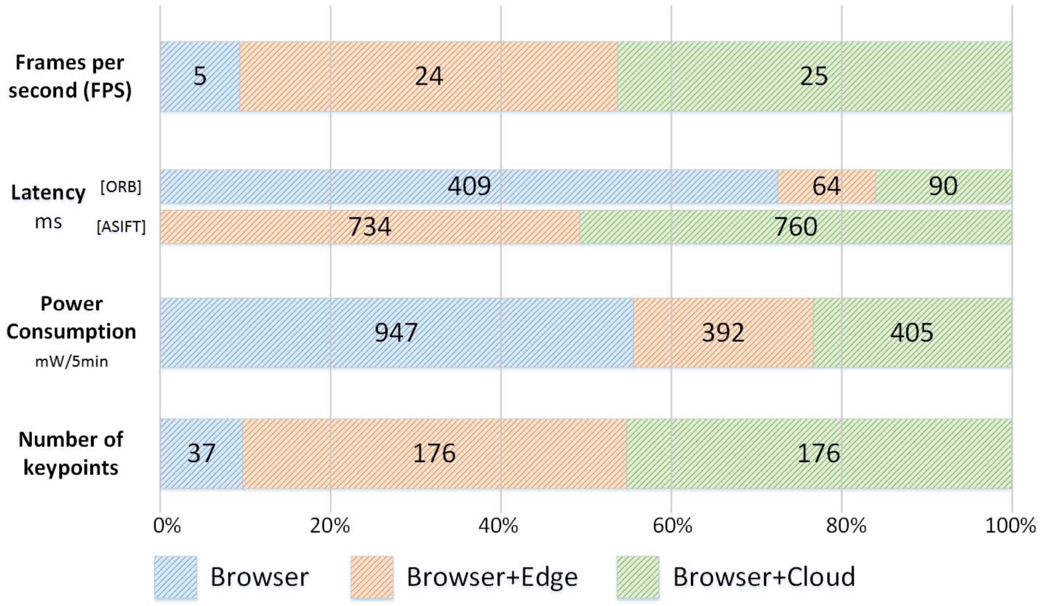

Figure 3. Experiment results in terms of frames per second, latency, the terminal's power consumption, and the average number of matched keypoints.

The experiment results show that MEC indeed provides a new choice for the future deployment of MAR.

FUTURE DIRECTIONS

The practical projects and performance results have given us great inspiration for the future development and deployment of web AR applications. Next, we discuss some important research challenges and directions for web AR and MEC.

Future Directions for Web AR

Web AR is becoming more and more popular owing to its lightweight and cross-platform features. We argue that it will provide a promising future for MAR. However, there is still a lot of work to be done to further promote the application of web AR.

Computational Efficiency for CPU-Hungry Tasks on Mobile Web Browsers

The widely used JavaScript performs poorly for complex computation tasks, such as matrix computation and floating-point computation. Therefore, it is necessary to introduce a more efficient computation paradigm to web browsers to meet the computational-efficiency requirement. Although some advanced techniques such as WebAssembly and Web Workers are already supported in some mainstream web browsers such as Chrome, Firefox, and Safari, there is still room for improvement to achieve efficient computation for end devices.

Lack of Standardization

Browser-kernel-based extension solutions for web AR implementations can make full use of end devices' hardware resources to achieve better performance. We envision that this has more potential as a web AR solution. However, a variety of current browser-kernel-based solutions have serious compatibility problems. Different web AR applications can be used only by their dedicated browser, which significantly limits the popularization of web AR. In addition, during the development of web AR applications, we found that there also exist serious compatibility issues between Web3D rendering technologies (such as three.js) and Web3D models made by different modeling tools such as 3ds Max, Maya, and Blender. To promote web AR applications on a large scale, the issue of compatibility mentioned above will be one of the most critical problems to be solved in the future.

Network Constraints

MAR is heavily dependent on network latency and bandwidth. However, wireless networks have an adverse effect on the performance of web AR applications. Although the current 4G networks already have good performance capability, they still cannot meet the low-latency requirements of new applications such as AR and VR. Technologies such as software-defined networking (SDN), device-to-device (D2D) communication, and mobile crowdsourcing mechanisms[6] offer new ways for wireless-network resource optimization. However, there is still plenty of optimization room to further improve the performance of web AR applications.

Future Directions for MEC

Because MEC is a newly proposed computing paradigm, it offers many opportunities for research, development, and deployment. In this section, we discuss potential research directions for MEC under 3G, 4G, and 5G networks.

MEC in 3G and 4G Networks

MEC has been explored by many carriers, including China Mobile and Huawei. However, it is still at an initial stage. Under the current 3G and 4G networks, many problems and challenges regarding the application, development, and deployment of MEC still exist. For MAR applications, the severe delay issues under wireless networks greatly degrade their performance. Currently, the content delivery network (CDN) closest to users is used mainly to cache the static content resources and has no computing and application deployment functions. Therefore, it’s necessary to upgrade the existing CDN as much as possible to support the edge-computing paradigm.

MEC in 5G Networks

5G mobile networks have recently gained momentum in both academia and industry. Benefiting from the inherent features of 5G cellular networks, such as enhanced mobile broadband (eMBB), massive machine type communications (mMTC), and ultra-reliable low-latency communications (URLLC), 5G is expected to enable a host of new applications, such as the Internet of Things (IoT), self-driving cars, AR and VR, and so on. Moreover, MEC is also available in 5G cellular networks to provide computing and storage abilities at the edge of a mobile network.[7]

Compared with current 3G and 4G networks, the issues of delay and bandwidth occupancy get alleviated in 5G networks. The introduction of MEC further optimizes the utilization of network resources and improves the performance of applications such as MAR, since the requests can be responded to faster in edge computing than with remote cloud-computing centers. However, when, where, and how to deploy MEC servers to fully balance the benefits, cost, and efficiency will be an essential issue in 5G networks.

CONCLUSION

MAR provides an attractive visual experience to users. Web AR further offers a lightweight, cross-platform, and pervasive solution on the web. However, it faces more technical challenges in terms of QoE owing to the weak computational efficiency of JavaScript. The emerging MEC paradigm provides an opportunity for the performance improvement of computationally intensive applications such as web AR. In this article, we proposed an MEC-oriented web AR solution. Then, we discussed challenges and future directions for MEC-based web AR under 3G, 4G, and 5G networks. We hope this article can provide insight on future research efforts in the emerging research field of web AR with MEC.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under grants 61671081 and 61720106007, the Beijing Natural Science Foundation under grant 4172042, and the 111 project (B18008).

REFERENCES

1. M. Billinghurst, A. Clark, G. Lee et al., “A Survey of Augmented Reality,” Foundations and Trends R in Human–Computer Interaction, vol. 8, no. 2-3, 2015, pp. 73–272.
2. A. Ahmed and E. Ahmed, “A Survey on Mobile Edge Computing,” Proc. 10th IEEE International Conference on Intelligent Systems and Control (ISCO), 2016, pp. 1–8.
3. P. Mendigochea, “WebAR: Creating Augmented Reality Experiences on Smart Glasses and Mobile Device Browsers,” Proc. ACM SIGGRAPH Studio, 2017, pp. 10.
4. B. MacIntyre, A. Hill, H. Rouzati, M. Gandy, and B. Davidson, “The Argon AR Web Browser and Standards-based AR Application Environment,” Proc. 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2011, pp. 65–74.
5. W. S. Shi, S. Dustdar, ”The Promise of Edge Computing, “IEEE Computer Magazine, Vol. 29, No. 5, 2016, pp. 78-81.
6. D. Zhao, X. Y. Li, H. D. Ma, ”How to Crowdsource Tasks Truthfully Without Sacrificing Utility: Online Incentive Mechanisms with Budget Constraint, “Proc. INFOCOM 2014, pp. 1213-1221.
7. T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges,” IEEE Communications Magazine, vol. 55, no. 4, 2017, pp. 54–61.

Xiuquan Qiao is a full professor in the State Key Laboratory of Networking and Switching Technology at the Beijing University of Posts and Telecommunications. His main research interests focus on 5G networks, augmented reality, and services computing. Contact him at qiaoxq@bupt.edu.cn.

Pei Ren is a PhD candidate from the Beijing University of Posts and Telecommunications. His research interests lie in machine learning and augmented reality. Contact him at renpei@bupt.edu.cn.

Schahram Dustdar is a full professor of computer science (informatics) with a focus on Internet technologies and heads the Distributed Systems Group at TU Wien. He’s a member of Academia Europaea. Contact him at dustdar@infosys.tuwien.ac.at.

Junliang Chen is a professor at the Beijing University of Posts and Telecommunications. His research interests include services computing. He’s a member of the Chinese Academy of Sciences and Chinese Academy of Engineering. Contact him at chjl@bupt.edu.cn.