

Fog Computing for Augmented Reality: Trends, Challenges and Opportunities

Shaik Mohammed Salman*, Taufik Akbar Sitompul[†], Alessandro Vittorio Papadopoulos[‡] and Thomas Nolte[§]

*Mälardalen University, ABB AB, Västerås, Sweden, shaik.salman@se.abb.com

[†]Mälardalen University, CrossControl, Västerås, Sweden, taufik.akbar.sitompul@mdh.se

[‡]Mälardalen University, Västerås, Sweden alessandro.papadopoulos@mdh.se

[§]Mälardalen University, Västerås, Sweden, thomas.nolte@mdh.se

Abstract—Augmented reality applications are computationally intensive and have latency requirements in the range of 15–20 milliseconds. Fog computing addresses these requirements by providing on-demand computing capacity and lower latency by bringing the computational resources closer to the augmented reality devices. In this paper, we reviewed papers providing custom solutions for augmented reality using the fog architecture and identified that the ongoing research trends towards balancing quality-of-experience, energy, and latency for both single and collaborative multi-device augmented reality applications. Furthermore, some works also focus on providing architectures for fog-based augmented reality systems and also on the training of machine learning algorithms in the fog layers to improve user experience. Based on these findings, we provide some challenges and research directions that can facilitate the adoption of fog-based augmented reality systems.

Keywords-fog computing, edge computing, cloudlets, augmented reality, trends, challenges, opportunities

I. INTRODUCTION

Augmented reality (AR) systems enhance the view of the real world by overlaying context-specific information on top of the real world [1]. The advantages of using AR systems in industrial contexts have been investigated, such as in shipyards [2], heavy machinery [3], remote-assisted maintenance [4], and industrial human-robot collaboration [5]. However, several technical limitations, such as the limited computing capacity, restrict wider adoption of AR systems in industrial contexts [2], [6]. In order to meet higher processing demands, AR devices need to be equipped with more computational power, which also requires integration of power supply units, such as batteries, to power these computing resources. However, this approach could make AR devices ergonomically uncomfortable [7]. Therefore, there is a need of accommodating the demand for high-computing capacity, without making AR devices bulkier and heavier than what we have today. To address the aforementioned problem, one notable approach that has been proposed is to offload some computational tasks of the AR

applications to remote computing nodes, such as cloudlets, edge servers, etc [8]–[10].

AR provides an interesting use case for fog computing. For example, the accuracy of context identification in AR applications highly depends on the video frame resolution, where higher resolution also means larger size of data. However, the transmission latency increases along with the increase in the size of the transmitted data [11]. In addition to the requirement on latency and accuracy of information provided, AR applications should also consider the requirement on user perception, energy consumption, network bandwidth, and transmission quality [8].

Fog computing architecture is a promising paradigm that envisions the execution of tasks among hierarchical computing layers by optimally distributing them to meet the demands of applications [12]. For example, high-frequency tasks that require low latency can be executed near end devices, whilst tasks that perform big data analysis can be executed at the cloud layer. This hierarchical architecture enables the execution of applications with low latency, while simultaneously reducing the usage of network bandwidth.

The use of fog computing as a mean for fulfilling the requirement of high computing capacity in AR applications is often mentioned in prior literature, for example, in Yi et al. [13], Dastjerdi et al. [14], and Satyanarayanan [15]. As we expect this trend is going to continue in coming years, we are interested in investigating how fog computing is used for supporting AR applications.

A comprehensive review of fog computing architecture has been provided by Yousefpour et al. [16]. AR-related hardware and software tools have been reviewed by Fraga-Lamas et al. [2]. The need for AR specific network protocols was discussed by Braud et al. [17]. Off-loading schemes based on machine learning for mobile edge computing platforms have been addressed by Cao et al. [18]. Mobile edge computing architecture and off-loading schemes have been reviewed by Mach and Becvar [11], where they com-

prehensively addressed works that consider energy and computation trade-offs. Additionally, they also reviewed works that consider off-loading based on application latency requirements. However, the review did not explicitly consider AR applications that require trade-offs between quality of experience, energy consumption, and latency considerations.

This paper reviews how fog computing has been used to support AR applications. The main objective is to highlight research trends that focus primarily on balancing the quality of experience, energy, and latency constraints of AR applications, for both single and multi-device use cases based on the fog computing architecture. Based on this, we discuss some key challenges and research directions that can facilitate the adoption of fog-based augmented reality systems in industrial contexts.

II. METHOD

We followed a systematic literature review approach [19], similar to the one used in [20]. Many papers use the terms edge, mobile edge, cloudlets, and fog to describe the intermediate computing layer between the cloud and the end devices. For the sake of common understanding, in this paper, we used the definitions as provided by the Open Glossary of Edge Computing project¹. To find relevant papers, we used the following search string on Scopus², and the search was done according to the paper’s title:

```
("augmented reality" OR " AR ") AND
("fog" OR "cloudlet" OR "edge")
```

The search result provided us with 79 papers that fit the search string above. There was no time limitation when we searched for relevant papers, thus all papers that were published up to October 2019 were included in our search. After reading the abstracts, we then reduced the number of papers by the following exclusion criteria:

- The term “edge” is used in the context of computer vision algorithms.
- The term “cloud” is used in the context of point clouds visualization.
- The term “AR” is used as an abbreviation of a mathematical model called “autoregressive”.
- The term “Ar” is used as an abbreviation of a chemical element “Argon”.
- The paper is a review paper.
- The term AR refers to augmented reality, but the approach is more relevant for virtual reality (VR), for example, off-loading computational tasks for generating 360-degree videos.

This process gave us 34 papers that we reviewed in detail.

¹<https://www.lfedge.org/projects/openglossary/>

²<https://www.scopus.com/search/form.uri?display=basic>

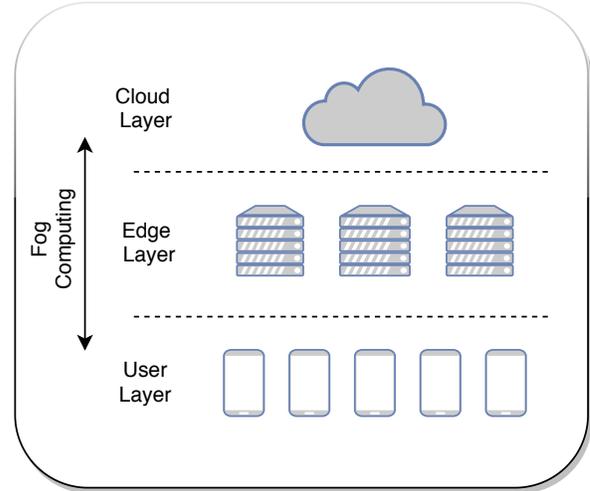


Figure 1. Layered Architecture.

III. TRENDS

AR applications require high computational capacity to provide seamless experience to users. Fog computing addresses this by enabling the availability of computing resources closer to users. However, executing all computational tasks in the edge layer introduces transmission delays while executing them locally requires high computing capacity on the device. We observed that much of the literature that investigates the use fog computing for AR applications is focusing on providing solutions that balance quality of experience (QoE) by trading-off between accuracy, latency, and energy consumption. Since both fog computing and AR are emerging technologies, a few papers also discussed architectures to provide fundamental support for fog-based AR applications. From an application perspective, fog computing has also been used to improve the AR experience by training AR-related machine learning algorithms in the edge layers. Lastly, some of the papers [21]–[28] evaluated the performance of edge-based solutions compared to cloud-based alternatives and demonstrated the advantages of edge-based solutions. Therefore, we classify the trends of this study under the following categories: (i) architecture, (ii) energy optimization, (iii) latency optimization, and (iv) applications.

A. Architecture

Fog-based applications are designed considering the hierarchical and layered fog computing architecture, as shown in Fig. 1. Here, the latency sensitive components of the application are executed closer to devices, while the layers closer to, and, including the cloud, are reserved mostly for data storage, monitoring, and co-ordination of the layers lower in the hierarchy. The papers reviewed in this study continue to follow this pattern.

Verbelen et al. [8] provided a component-based middleware for a cloudlet architecture and investigated the usefulness of such an architecture for AR applications. They considered resources such as laptops and mobile devices within the local network to be a cloudlet along with the computing resources in the cloud. They discussed the advantages of off-loading and the need for dynamic configuration of feature tracking points to match latency requirements. Bohez et al. [29] extended the architecture proposed by Verbelen et al. [8] and provided a middleware for collaborative AR applications. They extended the architecture by adding support for synchronization and shared off-loading. The synchronization allows maintaining a consistent application state across the collaborating devices while with shared off-loading, certain components that are common to all devices, are off-loaded to a cloudlet. The problem of optimal allocation of components between different resources was formulated as an optimization problem, where the objective function was to minimize the average CPU usage of all devices, including the cloudlets and the network bandwidth. They provided a simulated-annealing approach to solve this optimization problem. They evaluated the middleware and the algorithm on real hardware and showed that off-loading reduces the execution time between 41% and 95% when compared to the execution on local devices. Al-Shuwaili et al. [30] investigated a two-layered architecture that consists of cloudlets and end devices. To support a collaborative, multi-user AR application scenario, the cloudlets communicate with other cloudlets. Here, end devices are connected to a base station that is equipped with a cloudlet. However, the paper does not provide any further architecture-related information.

Schneider et al. [31] evaluated an edge-based architecture for augmented reality-based remote assistance. The architecture consisted of a client device, which can be any mobile device that has an integrated camera and an inertial measurement unit (IMU), and an edge resource, which is a high-performance computer. The client device simply captures the data from the camera and the IMU sensor and sends it to the edge resource. The edge resource computes the input data and return it back to the client device. They evaluated the proposed architecture and showed that although compressing the captured data required additional time for encoding and decoding, transmitting the compressed data resulted in lower latency than transmitting uncompressed data. All in all, the end-to-end latency was about 50 milliseconds. They pointed out that this latency was acceptable for hand-held devices, but too high for the head-mounted devices.

In contrast to the previous works, Zhou et al. [32] considered the case of supporting AR in vehicles and provide a vehicle-to-edge architecture framework for AR applications. Vehicles, environment, and edge resources are the main constituents of the architecture. They distribute the edge resources in two hierarchical layers. One is closest to the vehicles and is placed at base stations, while the other layer

is at “aggregation points” of the network. The architecture is designed to provide a real-time 3D visualization of the vehicle’s extended environment, for example, a 3D map of the road around or ahead of the vehicle. This map is constructed by combining data from several vehicles that transmit the data to the edge resources and is made available to new vehicles within the coverage area of edge resources. The latency is reduced, since only the map is provided and no other tasks normally associated with AR, such as tracking, are executed.

Fernández-Caramés et al. [9] proposed a three-layered architecture for AR applications customized for a shipyard. Here, end devices are parts of a local network connected via a wireless access point and they communicate with a so-called “local edge layer gateway”, which provides computing capacity to AR devices within the local network. Additionally, these edge layer gateways form a network among themselves to enable collaboration between remote AR devices. A cloudlet is also added to these gateway devices in the edge layer. Additionally, it is also assumed that it will provide more computational resources compared to the local gateways. Similar to the work by Fernández-Caramés et al. [9], Ren et al. [10] also proposed a three-layered architecture called “hierarchical computation architecture” that distributes different AR tasks between the end device and the edge. The architecture introduces “operation platform”, “virtualized controller”, and “communication unit” as components within the edge layer. The “operation platform” is responsible for processing AR tasks, whilst the “virtualized controller” acts as a coordinator for the complete edge layer. The “communication unit” is responsible managing communication within the edge layer resources, and communication with the cloud and the end devices.

Trinelli et al. [33] present a framework that enables a flexible and efficient way to process video streams for transparent acceleration of the AR application tasks within a network. The idea here is to make effective use of the network resources to accelerate the computation of AR workloads during data transmission in the network.

B. Energy Optimization

For wearable AR devices, fog computing provides a convenient opportunity to off-load parts of these computationally demanding tasks, thus reducing their energy consumption. Two of the reviewed papers [30], [34] explicitly considered energy as a parameter for optimization when deciding to off-load tasks, while two other papers [10], [35] only show that the energy consumption of AR devices is reduced compared to when the computation is done locally and in the cloud.

Al-Shuwaili and Simeone [30] proposed a solution for energy optimization when multiple AR devices are interacting with a shared application. The idea is, by utilizing mobile edge computing, each device involved in collaborative AR

applications does not need to send all data by itself. Instead, each device simply needs to send a fraction of relevant data to the cloudlet, and then the cloudlet simply returns a fraction of data that is relevant to specific devices only. The result shows that, by using shared cloud processing, the energy consumption was reduced 37% compared to separate off-loading. This result was obtained due to the shorter execution time and transmission period, thus lowering the energy consumption. Additionally, using shared uplink, 50% reduction was achieved, since each device only needs to transmit a fraction of its data. Altogether, the entire approach produces 63% energy reduction compared to separate off-loading.

Chatzieftheriou et al. [34] formulated an optimization problem to maximize the accuracy of context identification such as the detection of an object. The optimization problem used energy limits as one of the constraints. They showed that low latency requirements results in reduced accuracy, while energy constraints do not have significant impact on accuracy, i.e., allowing context identification tasks to run locally did not improve the accuracy of context identification.

C. Latency Optimization

The recommended frame rate for AR applications in head-mounted devices is 60 fps, which means it requires the latency to be less than 17 milliseconds³. Another recommendation is to have a consistent frame rate, for example, 30 fps over the complete duration is much better than varying frame rates within short intervals. There are eight papers that focus on optimizing the latency in fog-based AR applications. One distinct approach was to utilize a combination of techniques such as pipelined and parallel execution of different tasks to minimize latency. The remaining papers formulated an optimization problem to minimize latency. Therefore, we classify latency optimization under three themes: (i) latency as a constraint, (ii) latency as minimization function, and (iii) combined techniques.

1) *Latency as a Constraint*: Schneider et al. [31] investigated off-loading computing from AR devices to an edge server, called the “edge cloud” for a remote support application. Although the proposed approach introduces transmission delay between the client device and the edge cloud, compared to local execution, they showed that the time saved from off-loading the computing to the edge is still higher than the transmission delay. Moreover, compressing data requires more time for both encoding and decoding the data, but the results showed that transmitting this compressed data still produced lower latency than transmitting uncompressed data. All in all, for sending, tracking, annotating, and receiving a 752x480 resolution video compressed according to JPEG, the end-to-end latency was about 50 milliseconds.

They concluded that this latency is acceptable for a seamless AR experience using hand-held devices, but it is too high for a head-mounted devices.

Liu et al. [36] investigated off-loading computing from AR devices to an edge server, which could enhance AR devices’ capability to perform “fast and accurate object analysis”. The result showed that the performance of the object analysis is decreasing if the number of client devices is increasing. They compared the results of their approach with two other algorithms: (1) least connection and (2) random selection. They showed that their approach produced lower latency and higher accuracy of object analysis compared to the other two algorithms.

Liu and Han [37] proposed a “dynamic adaptive AR over the edge” protocol that adjusts the quality of augmentation (QoA) depending on the latency in different end devices. QoA is defined as a measure to evaluate the average accuracy of the object detection of edge-based AR applications. The protocol takes into account the relationship between video frame size, QoA, and latency. Based on this, they formulated an optimization problem that aims to maximize the QoA, while considering latency as a constraint in the edge layer. A custom algorithm based on the cyclic block coordinate gradient projection [38] is provided to solve the problem. The algorithm determines the image compression factor for the video frame size, along with the computation model for object identification and the resources on the edge. Based on the optimization results, they also provided a solution for virtual GPU (vGPU) resource allocation. The algorithm maps a user to a vGPU instance using a greedy approach.

Chatzieftheriou et al. [34] formulated an optimization problem to maximize the “precision of the item classification” for improved quality of experience by considering latency as one of the constraints. Although “item classification” is used in the general sense, here, it refers to object detection accuracy. The authors did not provide any solution for the optimization problem, but mention that any convex optimization solver should be able to provide an optimal solution. The evaluation showed that to have low latency, the trade-off is the precision of the classification.

Jia and Liang [39] investigated a multiplayer AR game scenario in an edge-based architecture. The players are divided into regions, with each region served by cloudlets within the region and a regional coordinator, that interacts with other coordinators to maintain a consistent state for all players within the game. Here, they focus on minimizing the universal “game frame duration”, which can be generalized as the latency, including communication between regional coordinators. To achieve this, they provided an iterative algorithm, that decides if AR computational tasks should be off-loaded to cloudlets and if yes, determining which of the participating cloudlets based on the connection strength and network bandwidth. To evaluate the algorithm, a simulation approach was used and the results showed that the

³<https://docs.microsoft.com/en-us/windows/mixed-reality/hologram-stability>

minimum latency is over 100 milliseconds and is linearly increasing with increasing number of players. It flattens as the player count increases over 1000 to roughly around 200 milliseconds.

Liu et al. [40] investigated latency minimization for shared data applications in the multi-user scenario. They formulated an optimization problem to minimize weighted latency of all devices in the collaborative application scenario under communication and computation constraints. Using task segmentation and joint resource allocation, the problem is rewritten as a convex problem and solved using Karush-Kuhn-Tucker conditions. The simulation results showed that the average latency per end device is over 20 milliseconds when the number of end devices is more than 20.

Liu et al. [41] consider “service failure probability” along with latency. “Service failure probability” is a combination of communication and computation failure probabilities along with timeout probability. The authors model computation failure as a Poisson process [42]. Communication error probability of transferring data between resources is modeled using block error rate and “uplink and downlink transport block” size of edge resources. The timeout probability refers to the latency. They proposed heuristic algorithms to solve the optimization problem of task off-loading defined using “service failure probability” as the minimization parameter and latency as the constraint. To address this, a heuristic algorithm is provided. The heuristic algorithm works by considering the AR tasks as a tree structured graph and divides the graph into smaller groups until the latency constraints are satisfied. The authors also provide an algorithm to dynamically select the data transmission rate and the offloading server based on Lyapunov optimization for wireless communication [43].

2) *Minimization Function*: Zhang et al. [44] considered the case of a collaborative AR application involving multiple AR devices. For such applications, they investigated co-located encoding and rendering as well as split encoding and rendering of the augmented video for transmission and display. To address both the task assignment and the off-loading problem, they formulated a multi-objective problem using the weighted sum method, which aims to minimize latency and maximize video quality. They solved this non-linear integer programming problem using a block coordinate descent method based algorithm. The evaluation showed the trade-off between latency, video quality and number of concurrent users, which is similar to the evaluation provided by Liu and Han [37].

3) *Combined Techniques*: Liu et al. [45] proposed a combination of techniques to minimize as well as to hide latency in order to achieve 60 fps. The techniques involve a “dynamic region of interest (RoI) encoding” scheme that reduces the bandwidth consumption by decreasing the quality of encoding of certain regions in the frame that may not necessarily contain any useful information. This is

coupled with a “parallel streaming and inference” technique, where inference begins on “slices of a frame”. Here, parts of the frame, referred to as slices, are encoded, transmitted, decoded, and inferred in a pipelined and parallel manner. The proposed inference mechanism called the “Dependency Aware Inference”, works on slices to provide high-accuracy object detection. Since encoding, transmission, decoding and inference simultaneously occur at different resources, i.e., encoding and transmission are carried out on the end device, while decoding and inference tasks are executed on the edge resources. This parallel approach reduces the end-to-end latency. Moreover, to improve the accuracy of object tracking and to hide the off-loaded latency, they also proposed a tracking technique called “motion vectors based object tracking”, which estimates the position of the object locally. Additionally, an adaptive off-loading mechanism was also proposed to determine which frames that should be off-loaded.

D. Applications

While papers that have been reviewed so far focused on codes off-loading, as well as energy and latency optimization, there are few papers that focus on the use of fog computing as a platform for training machine learning algorithms.

Ahn et al. [46] use fog computing for training a machine learning algorithm based on reinforcement learning. Due to the limited computing capacity, AR devices cannot be used for reinforcement learning. Instead, fog computing is used for training the algorithm and collecting environmental information that might be necessary for AR systems. Therefore, the AR system simply retrieves information computed by fog nodes. The reinforcement learning is used to prevent AR applications showing things that might be harmful for the user by using adaptive policies, for example, virtual objects should not be shown in a way that occludes important objects.

The user evaluation that was conducted in a virtual-reality environment showed that, by using the adaptive policies, where occluding objects are automatically made transparent or moved away from important objects, the AR application still generates adequate frame rates that support seamless experience. The data from the user evaluation also suggested that the adaptive policies still generate images that are good enough from the human perspective. However, the information regarding the fog computing itself is not mentioned in the paper.

In another paper, Ahn et al. [47] investigated the use of fog nodes as a platform for training imitation learning algorithm, which is another machine learning technique. The imitation learning was used for supporting personalization in AR applications, which learns about users’ preferences on where overlaid objects should be displayed on the environment.

The imitation learning consists of two parts: (i) teacher agent, which is controlled by the user, and (ii) student agent, which automatically captures the data from the teacher agent. After several training sessions that were conducted in a virtual-reality environment, the agent is able to learn users' preferred position of overlaid objects and where the physical trajectory where it was taken. In addition, both accuracy and precision are also improved along as the number of training sessions is increasing.

IV. CHALLENGES AND OPPORTUNITIES

In this section, we describe some challenges that we found based on the reviewed papers. At the same time, these challenges can also be seen as opportunities, where future work can be done for addressing the challenges.

Latency Requirements: As highlighted earlier, for a comfortable visual experience using head mounted displays, the frame rate should be around 60 fps. Analysis of the evaluation results of most of the papers shows that the latencies are greater than the 16.67 milliseconds even for low resolution video transmission when using the edge-based approach.

Benchmarks: Some of the papers reviewed in this study used simulation [41] and synthetic data sets [37] to evaluate their algorithms performance. This is sufficient to demonstrate the applicability of the algorithms when comparing them to cloud or local processing scenarios. However, since the evaluation is based on custom data sets, it is not possible to comment on the optimality of the solutions when compared to each other. Additionally, very few papers provide any details on the complexity of their proposed algorithms.

Security Aspects: AR devices transmit the data of users surroundings to the edge for processing. This information can be confidential or may need to be protected for privacy reasons. This requires some form of data encryption. Additionally, the reliability of the information provided to the user should be non-negotiable, especially when AR applications are used in the context of industrial systems [48]. However, the reviewed papers do not explicitly mention if these security aspects have been considered in their solutions nor is there any indication of their implicit consideration.

Resource Availability and Scheduling: AR applications have stringent timing requirements. The reviewed papers assume that the computing capacity at edge layers is reserved for each AR device and is available for use as soon as requested. However, this may not necessarily be true and the overhead of the task scheduling and blocking of resources by other tasks executing on the shared resources can have considerable influence on the latency of AR applications. To guarantee the timing requirements of AR devices, it can be useful to consider the use of real-time scheduling mechanisms in the edge layers [49], [50].

V. CONCLUSION

Fog computing enables execution of low latency and computationally demanding applications, such as AR, by provisioning resources closer to the user. In this study, we reviewed the literature to identify the research trends supporting the execution of AR applications in the fog computing paradigm. The review shows that the focus is primarily on managing QoE of AR applications by carefully considering the trade-off between accuracy, energy, and latency through formulation of optimization problems and finding solutions that meet the QoE requirements. Furthermore, we highlighted some of the challenges and opportunities that need to be addressed to enable the wider adoption of fog-based AR solutions.

VI. ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreements No. 764785, FORA—Fog Computing for Robotics and Industrial Automation, and No.764951, ImmerSAFE—Immersive Visual Technologies for Safety-critical Applications, and by the Swedish Foundation for Strategic Research under the project "Future factories in the cloud (FiC)", and the Knowledge Foundation (KKS) project SACSys.

REFERENCES

- [1] T. P. Caudell and D. W. Mizell, "Augmented reality: an application of heads-up display technology to manual manufacturing processes," in *Proc. 25th Hawaii International Conference on System Sciences*. IEEE, 1992, pp. 659–669 vol.2.
- [2] P. Fraga-Lamas, T. M. Fernández-Caramés, Ó. Blanco-Novoa, and M. Vilar-Montesinos, "A review on industrial augmented reality systems for the industry 4.0 shipyard," *IEEE Access*, vol. 6, pp. 13 358–13 375, 2018.
- [3] T. A. Sitompul and M. Wallmyr, "Using augmented reality to improve productivity and safety for heavy machinery operators: State of the art," in *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*. ACM, 2019, pp. 8:1–8:9.
- [4] R. Palmarini, J. A. Erkoyuncu, R. Roy, and H. Torabmostaedi, "A systematic review of augmented reality applications in maintenance," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 215 – 228, 2018.
- [5] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248 – 266, 2018.
- [6] M.-H. Stoltz, V. Giannikas, D. McFarlane, J. Strachan, J. Um, and R. Srinivasan, "Augmented reality in warehouse operations: Opportunities and barriers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12979 – 12984, 2017.

- [7] A. Martinetti, H. C. Marques, S. Singh, and L. van Dongen, "Reflections on the limited pervasiveness of augmented reality in industrial sectors," *Applied Sciences*, vol. 9, no. 16, 2019.
- [8] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "Adaptive deployment and configuration for mobile augmented reality in the cloudlet," *J. Network and Computer Applications*, vol. 41, pp. 206–216, 2014.
- [9] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and M. Vilar-Montesinos, "A fog computing and cloudlet based augmented reality system for the industry 4.0 shipyard," *Sensors*, vol. 18, no. 6, p. 1798, 2018.
- [10] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," *IEEE Network*, vol. 33, no. 4, pp. 162–169, 2019.
- [11] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [12] IEEE Standards Association, "IEEE 1934-2018 standard for adoption of openfog reference architecture for fog computing," 2018.
- [13] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. 2015 Workshop on Mobile Big Data*. ACM, 2015, pp. 37–42.
- [14] A. Dastjerdi, H. Gupta, R. Calheiros, S. Ghosh, and R. Buyya, "Chapter 4 - fog computing: principles, architectures, and applications," in *Internet of Things*, R. Buyya and A. V. Dastjerdi, Eds. Morgan Kaufmann, 2016, pp. 61 – 75.
- [15] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
- [16] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289 – 330, 2019.
- [17] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future networking challenges: The case of mobile augmented reality," in *37th International Conference on Distributed Computing Systems*. IEEE, 2017, pp. 1796–1807.
- [18] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 56–62, 2019.
- [19] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.
- [20] A. V. Papadopoulos, L. Versluis, A. Bauer, N. Herbst, J. von Kistowski, A. Ali-Eldin, C. L. Abad, J. N. Amaral, P. Tūma, and A. Iosup, "Methodological principles for reproducible performance evaluation in cloud computing," *IEEE Transactions on Software Engineering*, 2019.
- [21] R. Li, G. Gao, Y. Liang, X. Zhang, and Y. Liao, "An AR based edge maintenance architecture and maintenance knowledge push algorithm for communication networks," in *Proc. 2019 4th International Conference on Big Data and Computing*. ACM, 2019, p. 165–168.
- [22] A. F. Aljulayfi and K. Djemame, "Simulation of an augmented reality application for driverless cars in an edge computing environment," in *5th International Symposium on Innovation in Information and Communication Technology*. IEEE, 2018, pp. 1–8.
- [23] H. Yan and X. Qiao, "Research and implementation of edge computing in web AR," in *IOP Conference OPTseries: Materials Science and Engineering*, vol. 490. IOP Publishing, 2019.
- [24] S. Mai and Y. Liu, "Implementation of web AR applications with fog radio access networks based on openairinterface platform," in *5th International Conference on Control, Automation and Robotics*. IEEE, 2019, pp. 639–643.
- [25] C. Ling, M. Chen, W. Zhang, and F. Tian, "AR cloudlets for mobile computing," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 12, pp. 162–169, 2011.
- [26] X. Qiao, P. Ren, S. Dustdar, and J. Chen, "A new era for web AR with mobile edge computing," *IEEE Internet Computing*, vol. 22, no. 4, pp. 46–55, 2018.
- [27] H. Wang, J. Xie, and T. Han, "A smart service rebuilding scheme across cloudlets via mobile ar frame feature mapping," in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [28] Y. Liu, J. Ling, G. Shou, H. S. Seah, and Y. Hu, "Augmented reality based on the integration of mobile edge computing and fiber-wireless access networks," in *International Workshop on Advanced Image Technology (IWAIT) 2019*. SPIE, 2019, pp. 106 – 110.
- [29] S. Bohez, J. D. Turck, T. Verbelen, P. Simoens, and B. Dhoedt, "Mobile, collaborative augmented reality using cloudlets," in *International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*. IEEE, 2013, pp. 10–19.
- [30] A. N. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Letters*, vol. 6, no. 3, pp. 398–401, 2017.
- [31] M. Schneider, J. R. Rambach, and D. Stricker, "Augmented reality based on edge computing using the example of remote live support," in *International Conference on Industrial Technology (ICIT)*. IEEE, 2017, pp. 1277–1282.
- [32] P. Zhou, W. Zhang, T. Braud, P. Hui, and J. Kangasharju, "Enhanced augmented reality applications in vehicle-to-edge networks," in *22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2019.

- [33] M. Trinelli, M. Gallo, M. Rifai, and F. Pianese, "Transparent AR processing acceleration at the edge," in *Proc. 2nd International Workshop on Edge Systems, Analytics and Networking*. ACM, 2019, p. 30–35.
- [34] L. E. Chatzieftheriou, G. Iosifidis, I. Koutsopoulos, and D. J. Leith, "Towards resource-efficient wireless edge analytics for mobile augmented reality applications," in *15th International Symposium on Wireless Communication Systems*. IEEE, 2018, pp. 1–5.
- [35] 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 *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 349–350.
- [36] Q. Liu, S. Huang, and T. Han, "Fast and accurate object analysis at the edge for mobile augmented reality: demo," in *Proc. Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, pp. 33:1–33:2.
- [37] Q. Liu and T. Han, "DARE: dynamic adaptive mobile augmented reality with edge computing," in *26th International Conference on Network Protocols*. IEEE, 2018, pp. 1–11.
- [38] S. Bonettini, "Inexact block coordinate descent methods with application to non-negative matrix factorization," *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1431–1452, 2011.
- [39] M. Jia and W. Liang, "Delay-sensitive multiplayer augmented reality game planning in mobile edge computing," in *Proc. 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2018, pp. 147–154.
- [40] W. Liu, J. Ren, G. Huang, Y. He, and G. Yu, "Data offloading and sharing for latency minimization in augmented reality based on mobile-edge computing," in *88th Vehicular Technology Conference*. IEEE, 2018, pp. 1–5.
- [41] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11 222–11 236, 2019.
- [42] X. Qiu, Y. Dai, Y. Xiang, and L. Xing, "A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 3, pp. 401–412, 2016.
- [43] J. Liu and Q. Zhang, "Edge computing enabled mobile augmented reality with imperfect channel knowledge," in *European Wireless 2019; 25th European Wireless Conference*. VDE Verlag, 2019, pp. 1–6.
- [44] L. Zhang, A. Sun, R. Shea, J. Liu, and M. Zhang, "Rendering multi-party mobile augmented reality from edge," in *Proc. 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2019, p. 67–72.
- [45] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th Annual International Conference on Mobile Computing and Networking*. ACM, 2019.
- [46] S. Ahn, M. Gorlatova, P. Naghizadeh, M. Chiang, and P. Mittal, "Adaptive fog-based output security for augmented reality," in *Proc. 2018 Morning Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2018, pp. 1–6.
- [47] S. Ahn, M. Gorlatova, P. Naghizadeh, and M. Chiang, "Personalized augmented reality via fog-based imitation learning," in *Proc. Workshop on Fog Computing and the IoT*, 2019, pp. 11–15.
- [48] M. Langfinger, M. Schneider, D. Stricker, and H. D. Schotten, "Addressing security challenges in industrial augmented reality systems," in *IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 299–304.
- [49] R. I. Davis and A. Burns, "A survey of hard real-time scheduling for multiprocessor systems," *ACM Comput. Surv.*, vol. 43, no. 4, pp. 35:1–35:44, 2011.
- [50] S. Afshar, N. Khalilzad, F. Nemati, and T. Nolte, "Resource sharing among prioritized real-time applications on multiprocessors," *SIGBED Rev.*, vol. 12, no. 1, pp. 46–55, 2015.