UNIVERSITY OF HELSINKI

**https://helda.helsinki.fi**

**Helda**

# Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence

Cao, Jacky

Association for Computing Machinery
2023-09

Cao, J, Lam, K-Y, Lee, L-H, Liu, X, Hui, P & Su, X 2023, 'Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence', ACM Computing Surveys, vol. 55, no. 9. https://doi.org/10.1145/3557999

http://hdl.handle.net/10138/354524
10.1145/3557999

unspecified
acceptedVersion

*Downloaded from Helda, University of Helsinki institutional repository.*

*This is an electronic reprint of the original article.*

*This reprint may differ from the original in pagination and typographic detail.*

*Please cite the original version.*

# Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence

JACKY CAO, University of Oulu, Finland; Norwegian University of Science and Technology, Norway
KIT-YUNG LAM, The Hong Kong University of Science and Technology, Hong Kong
LIK-HANG LEE, KAIST, Republic of Korea
XIAOLI LIU, University of Helsinki, Finland
PAN HUI, The Hong Kong University of Science and Technology, Hong Kong; University of Helsinki, Finland
XIANG SU, Norwegian University of Science and Technology, Norway; University of Oulu, Finland

Mobile Augmented Reality (MAR) integrates computer-generated virtual objects with physical environments for mobile devices. MAR systems enable users to interact with MAR devices, such as smartphones and head-worn wearables, and perform seamless transitions from the physical world to a mixed world with digital entities. These MAR systems support user experiences using MAR devices to provide universal access to digital content. Over the past 20 years, several MAR systems have been developed, however, the studies and design of MAR frameworks have not yet been systematically reviewed from the perspective of user-centric design. This article presents the first effort of surveying existing MAR frameworks (count: 37) and further discuss the latest studies on MAR through a top-down approach: (1) MAR applications; (2) MAR visualisation techniques adaptive to user mobility and contexts; (3) systematic evaluation of MAR frameworks, including supported platforms and corresponding features such as tracking, feature extraction, and sensing capabilities; and (4) underlying machine learning approaches supporting intelligent operations within MAR systems. Finally, we summarise the development of emerging research fields and the current state-of-the-art, and discuss the important open challenges and possible theoretical and technical directions. This survey aims to benefit both researchers and MAR system developers alike.

CCS Concepts: • **Human-centered computing** → **Mixed / augmented reality**; *Graphical user interfaces*; • **Software and its engineering** → *Development frameworks and environments*; • **Computing methodologies** → *Machine learning approaches*.

Additional Key Words and Phrases: mobile augmented reality, user interactions, development framework, artificial intelligence, metaverse.

Just Accepted

## 1 INTRODUCTION

Over the past several decades, Augmented Reality (AR) has evolved from interactive infrastructures in indoor or fixed locations to various mobile devices for ubiquitous access to digital entities [135]. We consider the first research prototype of Mobile AR (MAR) to be the *touring machine system* [18] that offers road navigation cues on a school campus. After 20 years of development, MAR devices have shrunken from the size of huge backpacks to lightweight and head-worn devices. The interfaces on wearable head-worn computers have shifted from micro-interactions displaying swift (e.g., Google Glass) and small-volume contents to enriched holographic

Authors' addresses: Jacky Cao, University of Oulu, Oulu, Finland; and Norwegian University of Science and Technology, Norway, jacky.cao@oulu.fi; Kit-Yung Lam, The Hong Kong University of Science and Technology, Hong Kong, kylambd@connect.ust.hk; Lik-Hang Lee, KAIST, Daejeon, Republic of Korea, likhang.lee@kaist.ac.kr; Xiaoli Liu, University of Helsinki, Helsinki, Finland, xiaoli.liu@helsinki.fi; Pan Hui, The Hong Kong University of Science and Technology, Hong Kong; and University of Helsinki, Finland, pan.hui@helsinki.fi; Xiang Su, Norwegian University of Science and Technology, Norway; and University of Oulu, Finland, xiang.su@ntnu.no.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2022 Copyright held by the owner/author(s).
0360-0300/2022/8-ART
https://doi.org/10.1145/3557999

environments (e.g., Microsoft HoloLens) [133, 136]. MAR offers interactive user experiences on mobile devices by overlaying digital contents, e.g., computer-generated texts, images, and audio, on top of physical environments. Numerous MAR frameworks have been developed to support displaying such enriched real-world environments by managing different sensors and components for tracking physical objects and various user interaction features (e.g., anchors and ray-casting), as well as offloading computationally demanding tasks to remote servers. Nowadays, MAR, albeit primarily on smartphones, has been employed in various sectors, for instance, facilitating teaching and learning [113, 141, 148, 158, 223], surgery training and reducing human errors during surgical operations [34, 122, 138], virtual tour guides during sight-seeing [67, 73, 86, 271], product visualisation [9, 196, 270], navigating indoor and outdoor areas [72, 208, 282], supporting industry manufacturing [68, 227], and aiding with disaster management [42, 155, 179]. The multitude of MAR applications has proven its practicality in becoming a ubiquitous interface type in the real world. With the recent advancements of Artificial Intelligence (AI) and Internet of Things (IoT) devices, numerous mobile equipment and IoT devices construct smart and responsive environments. Users leverage head-worn computers to access a multitude of intelligent services through MAR. Accordingly, we advocate that MAR has to develop adaptability, or *adaptive MAR frameworks*, to address the on-demand user interactions with various IoT devices in the aforementioned smart environments, such as drones, robotics, smart homes, and sensor networks.

One critical research challenge to realise the MAR vision is providing seamless interactions and supporting end-users to access various MAR services conveniently, for instance, through wearable head-worn computers and head-mounted displays (HMDs). HMDs are technological enablers for user interactions within augmented environments. These headsets project digital contents in the form of windows, icons, or more complex 3D objects (e.g., virtual agents/avatars) on top of views of the physical world. Users with such wearable head-worn computers can access various AR services ubiquitously, and new forms of services and user experiences will spread and penetrate our daily routines. AR-enabled scenarios could include purchasing tickets from transport service providers, setting privacy preferences and control on small-size smart cameras and speakers, and buying snacks and drinks from convenience stores. The interfaces of MAR experiences are therefore ubiquitous service points to end-users.

AI is crucial to support features of high awareness of users' physical surroundings and context to support MAR experiences and achieve user-oriented services. Considerations of user performance, user acceptance, and qualitative feedback in such MAR systems make adaptive MAR user interfaces (UIs) necessary in rich yet complex physical environments. Several frameworks are available to support the creation of MAR experiences, where developers can deploy pre-built systems, which already contain features, such as displaying augmented objects, performing environmental analysis, and supporting collaborative AR with multiple users. From environment analysis found in MAR frameworks to adaptive MAR UIs, and to enable low-latency and good user experiences, AI methods are generally required to execute these tasks. Nowadays, more commonly, machine learning (ML) methods are typically used due to their ability to analyse and extract several layers of information from data (e.g., geographical data, user data, image frames taken from physical environments) in an efficient manner.

Thus, this survey examines the recent works of MAR, focusing on developments and synthesis of MAR applications, visualisations, interfaces, frameworks, and applied-ML for MAR. We also strive to move beyond the individual applications of MAR and seek research efforts towards highly intelligent and user-oriented MAR frameworks that potentially connect with not only digital and virtual entities but also physical daily objects. We outline key challenges and major research opportunities for user-oriented MAR UIs and the AI methods for supporting such UIs. The contributions of this survey paper are as follows:

(1) Provisioning a comprehensive review of the latest AR applications and framework-supported interface designs in AR in both hand-held and head-worn AR system scenarios.

Just Accepted

(2) Provisioning a comprehensive review of existing AR frameworks and SDKs, which explores the supported hardware platforms and features of the MAR frameworks.
(3) Provisioning a comprehensive review of relevant ML methods for application in key AR components of MAR frameworks.
(4) A research agenda for developing future adaptive AR user interactions concerning systematic MAR frameworks and ML-assisted pipelines.

Among our main calls to action in the agenda are issues related to AR experiences, such as investigating the feasibility of AR interfaces in user-orientated and highly adaptive manners, developing ML methods for MAR experiences in dynamic and complicated environments, and advancing MAR systems for two-way and seamless interactions with all intelligent objects in evolving MAR landscapes.

## 1.1 Overview of Mobile Augmented Reality

The basic principle of AR is that virtual objects are overlaid onto physical environments. These objects are dynamic digital assets rendered by user devices and may include 3D models, 2D videos, and 2D text labels. The objects' augmentation on the real world can be achieved in different ways. One method is to embed the objects as if they were part of the environment. This embedding is performable if a mapping of the users' surroundings is already pre-collected, e.g., with SLAM (see Section 4) [187]. An alternative method is to track with respect to the user and the device they are using for AR [289]. In this way, virtual content could always be in view, regardless of where they are looking or where the device is pointing.

Figure 1 presents the typical pipeline of a MAR system. The components in MAR systems are approximately groupable into *MAR Device* and *MAR Tasks*, where devices or external servers may execute the tasks. Generally, hardware sensors (i.e., cameras, GPS and IMU sensors, and LiDAR modules) on the MAR device first capture images and sensor data, which are then fed into the MAR tasks. The first is frame pre-processing, where data is cleaned and prepared to be inputted into the analysis tasks to retrieve information on which objects are in the environment, e.g., object detection, feature extraction, object recognition, and template matching tasks. These tasks are classifiable as the *ML Tasks*, where ML techniques help complete the tasks more efficiently due to their ability to learn from previous problems. The latter task of object recognition typically uses a database containing objects to compare the input. This task is to find potential matching objects, their labels, and other related information. After data analysis, the successful results and associated annotations are returned to the MAR device for object tracking and annotation rendering on the client. The virtual annotations and objects are overlaid on either a camera feed of the environment or rendered directly on users' views of the physical world, depending on the type of MAR device in use. Together with ML methods, this can form an overall *AR Framework* where developers can employ pre-built frameworks to create their own MAR applications. The annotation rendering and display components of the MAR device are grouped as part of *Adaptive UI*. The contents and interaction with the rendered augmentations are essential for users, allowing them to experience MAR. Therefore, the scope of this survey primarily aims to address the following questions which are missing in the existing surveys (detailed in Section 1.2): (1) What features and functions are available in MAR frameworks nowadays? (2) What are the latest efforts of AR research and, accordingly, the prominent gap between these efforts and the identified frameworks? (3) Then, based on the above questions, how do MAR engineers and practitioners design the next generation of MAR frameworks that bridge the user requirements?

## 1.2 Related Surveys and Selection of Articles

This subsection briefly outlines previous relevant surveys in various domains, i.e., user interfaces, frameworks, and intelligence. Papagiannakis *et al.* [184] present one of the earliest surveys focusing on different types of mobile and wireless technologies and their impact on AR, including computing hardware, software architecture,

Just Accepted

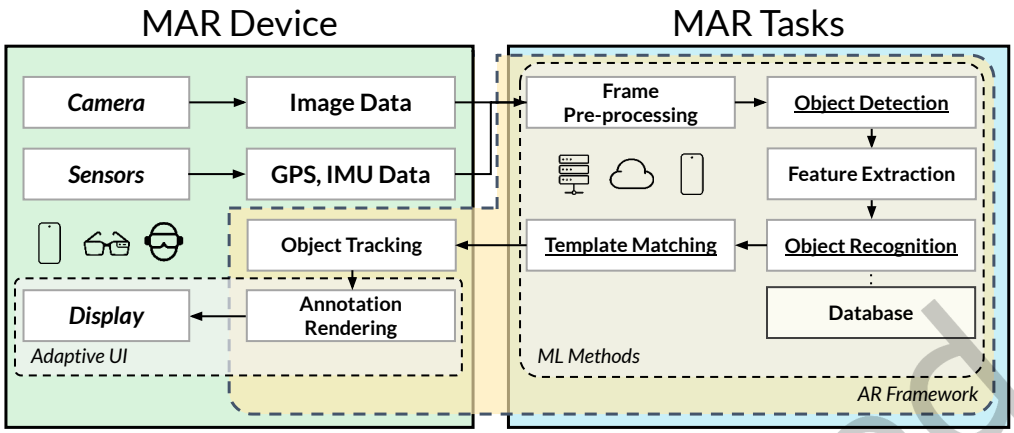


Fig. 1. A typical MAR system pipeline, splitting into MAR Device and MAR Tasks. ML methods can be utilised in the majority of required MAR tasks.

Just Accepted

wireless networking, tracking, and registration. Another early AR survey, focusing on AR systems, applications, and supporting technologies, is proposed by Carmigniani *et al.* [45]. Later on, Lee and Hui [135] focus on the user interaction techniques on MAR devices (i.e., smartglasses). A recent survey by Chatzopoulos *et al.* [46] present a comprehensive state-of-the-art survey of MAR application fields, how MAR user interfaces can be developed, what key components MAR systems consist of, and how data is managed in MAR systems. Zollmann *et al.* [296] re-organise the recent works of different components in the visualisation pipelines of AR systems and presents the findings in three main areas, including filtering, mapping, and rendering. Marneanu *et al.* [156] provide an early evaluation of AR frameworks for Android developments. Recent AR development platforms on Android and iOS have been discussed in some development communities but have never been evaluated systematically and scientifically. Herpich *et al.* [95] perform a comparative analysis of AR frameworks, albeit for the scenario of developing educational applications. Grubert *et al.* [85] discuss the context awareness in pervasive AR towards the dimension of graphics and content visualisation but lack the connection with ML. Braud *et al.* [39] present a review of the existing network infrastructures and protocols and propose several guidelines for future real-time and multimedia transport protocols, focusing on MAR offloading. Siriwardhana *et al.* [229] recently provided a comprehensive survey on the landscape of MAR with respect to 5G systems and multi-access edge computing. The authors further discuss the requirements and limitations of technical aspects, such as communication, mobility management, energy management, service offloading and migration, security, and privacy, with analysing the role of 5G technologies. However, these two papers focus on communications and networking perspectives. Finally, the utilisation of ML in different core components of MAR applications is discussed in several surveys. For example, Krebs *et al.* [129] present a survey on leveraging deep neural networks for object tracking. Liu *et al.* [144] focus on deep learning for object detection. Yilmaz *et al.* [285] survey object tracking. Sahu *et al.* [220] conduct a comprehensive study on fundamental ML techniques for computational pipelines of AR systems (e.g., camera calibration, detection, tracking, camera pose estimation, rendering, virtual object creation, and registration), in AR-assisted manufacturing scenarios. Our article addresses the gaps in these existing surveys and provides a comprehensive survey on MAR interface design, frameworks and SDKs, and appropriate ML methods utilised in key AR components.

The articles are initially selected by searching recent research articles using a small number of keyword seeds on Google Scholar, IEEE Xplore, ACM Digital Library, Scopus, Web of Knowledge, etc. The queries combine keywords from the set *MAR, ubiquitous interaction, user interaction, AR adaptive UI, AR visualisation techniques, AR camera calibration, camera pose estimation, AR remote collaboration, AR user collaboration, MAR frameworks, object detection, object tracking, object recognition, rendering, etc.* The initial article set was expanded by taking into consideration articles that cite or are cited by the articles within this set. We further complement the articles by adding prominent research featured in media to cover a comprehensive set of important publications in this area. This process was continued until no new articles were found. We then discussed the papers among the authors and analysed the most relevant and important selected articles by reading the abstract and the main findings of the papers. Papers that were less relevant to the scope of the survey were filtered out during this process. The selected papers form the core of this survey, and we have performed continuous updates during the survey writing process to cover papers published since the start of our process.

### 1.3 Structure of the survey

The remainder of this article is organised as follows. Section 2 summarises significant application areas for MAR. We present detailed discussion and analysis on the development and recent advances of adaptive UI for MAR, MAR frameworks, and ML for MAR, in Section 3, Section 4, and Section 5, respectively. Finally, we discuss research challenges and future directions in Section 6, and conclude the paper in Section 7.

## 2 MAR APPLICATIONS

Over the past few years, AR has gained widespread media attention and recognition due to easy access through modern smartphone applications. As an emerging technology, research and development of MAR have been seen in several different fields. The versatility and ease of use of MAR highlight the potential for the technology to become a ubiquitous part of everyday life. In 2017 alone, the revenue of the AR market was worth $3.5 billion [145] and by 2023 this value could reach up to $70–$75 billion [163]. A study conducted by SuperData and Accenture Interactive found that 77% of AR users are likely to use AR as a content viewer (e.g., for file viewing), 75% are likely to use AR in shopping scenarios, and 67% are likely to use AR for gaming entertainment. This reveals the importance of considering AR user interfaces, the underlying AR system, and how users can continually be encouraged to use AR. However, the user interface requirements for each of the various AR scenarios are varied. In this section, we study several particular AR scenarios and applications, emphasising the types of user interfaces employed. This leads to a greater understanding of why AR has been such a prominent technology and why it continues to attract research attention.

### 2.1 Entertainment

AR technologies push the boundaries of entertainment and tourism. Pokémon GO [211] is a mobile game that uses AR to bring fictional Pokémon characters into the real world. Released in July 2016, at the height of the game's popularity, there were 45 million users and generated a first month of revenue of $207 million [106]. This has led to the creation of similar concept AR gaming apps, such as Pikimin Bloom [177]. AR gaming is not restricted to enhancing optical camera feeds. Audio AR experiences are another type of augmentation, such as in the running app, Zombies, Run! [230]. Social media applications, such as Snapchat and Instagram, allow users to apply "face filters" or "lenses" to themselves using AR [232]. In addition, Snapchat and social media application TikTok have utilised the LiDAR sensor in the 2020 generation of iPhones to create environmental AR effects [76].

Just Accepted

## 2.2 Tourism

AR could be digital tour guides for landmarks or cultural sites in tourism applications. Several projects create interesting MAR applications, which are offered to visitors to Tuscany, Italy [67] and Basel, Switzerland [73] with city navigation guidance, sight-seeing destinations, and general information. The design of such applications is traveller-centric, so information is tailored to specific audiences, for example, providing multi-language functionality [86], offering alternative haptic or audio AR experiences [271], or supplying information that is suited to an individual's knowledge level [128]. Widespread adoption of AR has yet to emerge. However, with continual exposure and improving technologies, there is no doubt that AR will become a prevalent part of everyday life in several tourism sectors. Some popular AR tourism applications, such as Horizon Explorer [20], provide travel information on top of the AR screen. AR frameworks, such as ARKit, are widely adopted in these applications to enable adaptive and relevant content for travellers.

## 2.3 Education

Over the past 30 years, classrooms have been evolving through various technologies. From computers in classrooms [60] to lecture theatres with projection systems [173], classroom-based learning is now incorporating AR technology as an educational tool. Educational AR can be categorised as subject-specific AR applications and separate AR teaching tools. Subject-specific AR applications are developed for promoting a specific set of knowledge, and the user interfaces for these applications are well designed for presenting interactive content. AR teaching tools apply AR technologies to improve the efficiency of teaching. The AR teaching tool user interface facilitates users' communications during educational activities. Advantages are shown to arise from the use of AR in educational settings. For example, the individual skills of students can be improved, such as an increased ability to understand and handle information [113, 223], enhanced confidence [148], and development in spatial abilities [121, 141, 158].

AR could enhance student-teacher communications within classrooms. Zarraonandia *et al.* [288] suggest providing teachers with HMDs and students with devices which can alert the teacher if a student is struggling. Holstein *et al.* [97] design intelligent tutoring systems as a rich formative assessment tool, which can augment teachers' perceptions of student learning and behaviour in real-time. For students with disabilities and learning difficulties, using AR can tailor the teaching to individual specific needs. For example, pupils with attention deficit hyperactive disorder could use AR to help emphasise and improve engagement, motivation, and content visualisation [2]. Martín-Gutiérrez *et al.* [157] present an AR tool that allows for interactive and autonomous studying and collaborative laboratory experiments with other students and without assistance from teachers. Besides indoor classrooms, MAR is also useful for outdoor teaching, such as scientific excursions [28, 235, 294]. Night Sky and skyORB [183] are popular AR astronomy applications available for iOS. Users can learn about the position of stars and planets from the AR overlay when holding their mobile devices towards the sky.

## 2.4 Healthcare

AR applications in the medical field offer new approaches to patient-doctor relationships, treatments, and medical education. For example, AR is categorised as an assisting technology to support individuals who have Alzheimer's disease to help identify objects and people, provide reminders to take the correct medication, and aid caregivers in locating patients [117]. Alternatively, current medical imaging modalities, such as ultrasound and CAT scans, can be significantly enhanced through scan virtualisation into 3D models and their superposition onto physical body parts [190]. In a scenario such as a pregnant mother receiving an ultrasound, the scan could be cast in real-time to several observers wearing HMDs in the same room. All participants could obtain the identical ultrasound hologram on top of the patient, but midwives and other medical staff could be viewing additional information, such as past medical history and other vital signs [49, 153]. Novarad OpenSight presents one particular Microsoft

Just Accepted

HoloLens application product for CT images [79], allowing a user to view virtual trajectory guides and CT images that are superimposed on physical objects as two or three-dimensional augmentations.

One of the most significant areas of AR for medical research is utilising AR for surgery in various stages of treatment. Firstly, surgery can be explored and planned before execution, which allows surgeons to develop the optimal treatment strategy [37]. Once the surgical procedure begins, image guidance can provide surgeons with navigation information, so their attention is not drawn away from the operative field [58, 70]. In addition, mixed reality image-guided surgery (IGS) has been frequently used for surgery training [34, 122] and minimally invasive surgeries. In IGS, surgical instruments are tracked and visualised with respect to patient-specific data sets to guide surgeons while in the operating theatre. Several surgical fields have already begun to apply AR, for example, neurosurgery, general surgery, and orthopaedic surgery [228].

Adaptive user interface techniques are critical for surgery scenarios. Several techniques, including label placement, occlusion representation, and registration error adaptation, are extensively studied to support this. For example, Fuchs *et al.* [70] use depth-cue occlusion to guide the depth perception of multiple objects placement in laparoscopic surgery. In the AR system proposed by Wieczorek *et al.* [274], occlusion handling of the surgeon's hands gives a more realistic perception and blending of realities. Pratt *et al.* [200] demonstrate that AR can assist accurate identification, dissection, and execution of attached vascular flaps during reconstructive surgery.

## 2.5 Marketing

AR is a helpful tool for commercial businesses selling and marketing products and services. The IKEA Place [103] application is one notable example, allowing users to place digital assets of IKEA furniture within their local environments and see whether the items fit in both decor style and physical location. For fashion and footwear, there are already different AR applications for users to virtually try items before purchasing them. This has advantages such as removing barriers to trying the latest fashion trends, allowing users to personalise products before ordering uniquely, and minimising hygiene issues. WANNA [270] is one application for trying sneakers in AR. Alongside their own application Wanna Kicks, their AR system is used in the applications of other fashion brands such as Gucci and Reebok, and the social media application Snapchat. ASOS is an online fashion retailer that also uses AR for product try-outs within their mobile shopping applications [196]. Alternatively, there are physical retail locations such as Amazon Salon, where AR is being used for digital hair consultations [9]. Similar to the other marketing strategies which use AR, consumers can try different styles before fully committing to purchasing the hair styling service.

## 2.6 Navigation

Map-based applications are the typical method used for navigating and reaching destinations. However, these applications rely on GPS to locate the user and provide instructions. While this works effectively outdoors with good GPS coverage, indoor navigation suffers as the signals are attenuated by buildings, walls, objects, etc. [204]. Therefore, AR is one solution where object recognition can be performed on the environment to localise the user and overlay navigation instructions on the environment. Fusco *et al.* [72] demonstrate such an indoor system by recognising informational signs, landmarks, and structural features to create a turn-by-turn indoor mobile navigation application. Wu *et al.* [282] present a similar system where object recognition is performed on numbered door signs to locate users within a building, and then AR annotations show directions. AR-based navigation systems are naturally not limited to just the indoors; the technology is equally applicable for outdoor navigation. Object recognition and GPS can be fused to create AR navigation applications [140, 267]. Live View [208] is a feature in the commercial Google Maps application, which uses these technologies to help users navigate indoor and outdoor environments. This is made possible by applying ML object recognition algorithms on Google Street View data [81]; the application can then show augmented instructions to the user.

Just Accepted

### 2.7 Industry

For industries, AR has the potential to impact and support the sector significantly. Considered as one of the pillars of Industry 4.0 [256], AR can help achieve the goal of entirely digitised and intelligent industries [63]. The scope of AR is not just limited to one specific method or area in industry; instead, AR could be used for several industrial services and processes, for example, in the design process [41, 169], and supporting manufacturing and construction [68, 227]. However, two areas in industry that could significantly exploit AR are training and maintenance. With training, required competencies and knowledge in specific tasks can be developed using visual augmentation technologies. Individuals may be given tailored step-by-step training executed on or near the job. This allows them to learn processes without the need to leave the work area to consult training documents [234] saving time and cost resources and improving safety and situational awareness by providing employees with alerts to potentially hazardous situations [154]. Similarly, for maintenance, AR could be used in the 3D visualisation of complex instructions, which may not be naturally intuitive when viewing from 2D technical manuals [225]. There could also be the possibility that an individual is unable to perform maintenance due to a lack of technical knowledge or skill. In these scenarios, a support operator can provide remote assistance to the individual by delivering guided augmented instructions to their AR device [159, 168, 264].

### 2.8 Disaster Management

In the aftermath of a disaster, whether natural or otherwise, there is a need for urgent and prompt action to secure the safety of the people and area which has been affected. In addition to this, there need to be additional steps taken to mitigate any more potential harm and damage. AR is helpful in this regard, i.e., for disaster response and management, as AR's core aspect of augmenting information on the real world can be fully exploited. For example, Markus *et al.* [155] describe a concept for a disaster management tool that uses an AR user interface to help analyse the situation at rescue sites which leads to an acceleration in response activities. In an earthquake scenario, emergency teams could view information such as superimposed 3D models of undamaged buildings. They are then able to assess which areas could contain trapped victims. This assessment could be guided by a situationally aware management system that gathers information from sources such as users, sensors, and crowdsourcing [179]. In other emergency response situations, an AR tool could contain abilities to report points of interest and incidents for other first responders and a way to communicate between them to ensure that action is taken as soon as possible [42].

## 3 COMPOSING AR VISUALISATION AND USER ENVIRONMENTS

Among all the AR applications (Section 2), the general AR experience is to display virtual entities overlaid on top of the physical world. For producing immersive AR experiences, information management and rendering are key processes for delivering virtual entities. A collaborative AR system enables users to collaborate within the physical and virtual environments. Numerous MAR frameworks are designated for the purposes mentioned above. Before we describe the existing MAR frameworks (Section 4), this section provides an overview of the relevant concepts and current works in UI adaptability and collaborative UIs when there are multiple users and multiple devices. We also explain the existing metrics for measuring the effectiveness of AR.

### 3.1 Adaptive UI for MAR

After data is prepared for visualisation, there are three main steps in the visualisation pipeline [296], including (1) filtering, (2) mapping, and (3) rendering. Zollmann *et al.* [296] perform a comprehensive review organised by this visualisation pipeline. We adopt their approach for classifying the characteristics of visualisation techniques, and we further investigate the context-aware adaptive interface techniques which adapt to a person, task, or context environment. Several algorithms, such as ML-based algorithms, have been applied in adaptive user

Just Accepted

interfaces for improving human-interface interactions. They are designed to aid the user in reaching their goals efficiently, more easily, or to a higher level of satisfaction. We evaluate these works by considering the user experience measurement perspectives and summarise four adaptive user interface techniques [112, 296] commonly found in the existing frameworks, with the following requirements for the implementations of each set of interfaces:

**Information Filtering and Clustering (InfoF):** Overload and clutter occur when a large amount of information is rendered to users. Complex visualisation of augmented contents negatively affects visual search and other measures of visual performance. In order to produce better user interfaces, there are two approaches to reduce the complexity of interfaces by decreasing the amount of displayed content. The first approach is information filtering, which can be implemented by developing a filtering algorithm that utilises a culling step and a detailed refinement step. Another approach is information clustering [246], which groups and represents the information by their classification.

**Occlusion Representation and Depth Cues (OcclR):** Occlusion representation provides depth cues to determine the ordering of objects. Users can identify the 3D locations of other physical objects when large structures occlude over or under each other. Julier *et al.* [111] suggest three basic requirements for depth cues, including (1) the ability to identify and classify occluding contours; (2) the ability to calculate the level of occlusion of target objects and parameterising the occlusion levels if different parts of the object require this; and (3) the ability to use perceptually identified encodings to draw objects at different levels of occlusion.

**Illumination Estimation (IEsti):** Light estimation is another source of information depicting the spatial relations in depth perception, enhancing visual coherence in AR applications by providing accurate and temporally coherent estimates of actual illumination. The common methods for illumination estimation can be classified into auxiliary or non-auxiliary information.

**Registration Error Adaptation (RegEA):** Another mapping step of the rendering pipeline. Trackers are imprecise, and time-varying registration often exists. Therefore, the correct calibration of devices and displays is complex. This subsequently leads to graphical contents not always aligning perfectly with their physical counterparts. Accordingly, the UI should be able to adapt while visualising the information dynamically. Ambiguity arises when the virtual content is not interacting with the context environment around users.

**Adaptive Content Placement (AdaCP):** AR annotations are dynamic 3D objects which can be rendered on top of physical environments. Text labels must be drawn with respect to the visible part of each rendered object to avoid confusing and ambiguous interactions.

**InfoF**: Information clutter is prevalent in some context-aware AR applications due to complex outdoor environments. Exploring and searching for information on AR screens become indispensable tasks for users. Information filtering is a necessary technique in modern AR systems [249], especially in large and complicated environments, where information overload is significant. Without intelligent filtering and selection tools automation, the display would always lead to difficulty in reading information [123]. There are three main methods for filtering information, including (1) spatial filters, (2) knowledge-based filters, and (3) location-based filters [249]. Spatial filters select information displayed on screens or in the object space based on physical dimension rules. These filters require user interactions to investigate the entirety of the virtual content. For example, users must move their MAR devices to view large 3D models. But the method only works locally in a small region. The immersive AR application always applies spatial filters to exclude the information that is out of the user's view. Knowledge-based filters enable user preferences to be the filtering criteria [249]. Expert knowledge filters embed behaviour and knowledge in the coding, regulating the system's data structures to infer recommendations and output the items satisfying user requirements. Such knowledge coding can be done in different ways, such as rules in rule-based systems [195]. Finally, spatial information from the location-based filters can be combined with knowledge-based

Just Accepted

filters as hybrid methods. As new sensors are embedded in modern AR headsets, such as gaze sensors, the user bio-information and context information can be used for filtering [6].

**OcclR**: Comprehensive AR systems track sparse geometric features and compute depth maps for all pixels when visualising occluded objects or floating objects in AR [112]. Depth maps provide depth values for each pixel in captured scenes. They are essential for generating depth cues, helping users understand their environment, and aiding interactions with occluded or hidden objects. Recent AR frameworks, such as Google ARCore [82] and Apple ARKit [16], provide depth map data for enabling depth cue features in MAR applications [59]. Physical and virtual cues are two options for producing depth cues in AR applications to support depth perception [296]. Physical cues can be used to rebuild natural pictorial depth cues [278], such as occlusion or shadows. Integrating depth maps and RGB camera images can provide the necessary natural pictorial depth cues [296]. Subsequently, virtual cues and visual aids are generated by applications to provide similar depth cues as physical alternatives. "X-ray vision" is a technique for virtual cues frequently used to perceive graphics as located behind opaque surfaces. DepthLab [59] is an application using ARCore's Depth API, enabling both physical and virtual cues, helping application developers integrate depth into their AR experiences. DepthLab implements the depth map and depth cues for at least six kinds of interactions in AR, including (1) oriented reticles and splats, (2) ray-marching-based scene relighting, (3) depth visualisation and particles, (4) geometry-aware collisions, (5) 3D-anchored focus and aperture effect, and (6) occlusion and path planning [59]. Illumination estimation is typically achieved with two traditional approaches, including: (1) Methods utilising auxiliary information which leverages RGB-D data or information acquired from light probes, and the methods can be an active method like the fisheye camera used by Kán *et al.* [114] or a passive method like reflective spheres used by Debevec [57]. (2) Estimating the illumination using an image from the primary AR camera without the need of having an arbitrary known object in the scene. The auxiliary information can also be assumptions of some image features which are known to be directly affected by illumination or simpler models like Lambertian illumination. Shadows, the gradient of image brightness [119] and shading [273] are the typical image features for estimating illumination direction.

**RegEA**: Addressing registration and sensing errors is a fundamental problem in building effective AR systems [24]. Serious registration errors can produce conflicts between user visual inputs and actions, e.g., a stationary user is viewing AR content that appears to be moving away from the user at constant momentum. Such conflicts between different human senses may be a source of motion sickness. Therefore, the user interface must automatically adapt to changing registration errors. MacIntyre *et al.* [150] suggest using Level Of-Error (LOE) object filtering for different representations of augmentations to be automatically used as registration error changes. This approach requires the identification of a target object and a set of confusers [151]. Afterwards, their method calculates the registration errors for the target and all confusers. The delegation error convex hulls are used to bound the geometry of the objects. The hulls are constructed for two disjoint objects in the presence of substantial yaw error. The hull surrounding each object with a suitable label is sufficient to direct the user to the correct object edges.

Registration error adaptation is critical in safety-critical AR systems, such as for surgical or military applications. Recent AR frameworks provide real-time registration error adaption with precise IMU tracking data and camera image fusion algorithms, which minimises the registration error. Robertson and MacIntyre [217] describe AR visualisation techniques for augmentations that can adapt to changing registration errors. The first and traditional technique is the provision of a general visual context of augmentations in the physical world, helping users to realise the intended target of an augmentation. This is achieved by highlighting features of the parent object and showing more feature details as the registration error estimate increases. The second technique presents detailed visual relationships between augmentation and nearby objects in the physical world. A unique collection of objects near the target of the augmentation in the physical world is highlighted, and the user can differentiate between the augmentation target and similar parts of the physical world.

Just Accepted

**AdaCP**: Major label placement solutions include greedy algorithms, cluster-based methods, and screen subdivision methods [25]. Other methods include making the links between objects and their annotations more intuitive, alleviating the depth ambiguity problem, and maintaining depth separation [161]. Labels must be drawn with respect to each visible part of the object. Otherwise, the results are confusing, ambiguous, or even incorrect. By computing axially-aligned approximations of the object projections, the visibility is then determined with simple depth ordering algorithms [31]. Several research works focus on providing appropriate moving label dynamics to ensure that the temporal behaviour of the moving labels facilitates legibility. From these works, certain requirements arise, such as the ability to determine visible objects, the parameterisation of free and open spaces in the view plane to determine where and how content should be placed, and labels should be animated in real-time because the drawing characteristics should be updated on a per-frame basis [112]. The aforementioned requirements are the core of content placement that adapts to various physical environments for enhanced user experiences.

View management algorithms address label placement [25]. Wither *et al.* [276, 277] provide an in-depth taxonomy of annotations, especially regarding the location and permanence of annotations. Tatzgern *et al.* [249] propose a cluster hierarchy-based view management system. Labels are clustered to create a hierarchical representation of the data, which is visualised based on the user's 3D viewpoint. Their label placement employs the "hedgehog labelling" technique, which places annotations in real-world space to achieve stable layouts. McNamara *et al.* [162] illustrate an egocentric view-based management system that arranges and displays AR content based on user attention. Their solution uses a combination of screen position and eye-movement tracking to ensure that label placement does not become distracting.

For a comprehensive MAR system, current interfaces do not consider walking scenarios. Lages *et al.* [130] explore different information layout adaptation strategies in immersive AR environments. A desirable property of adaptation-based interface techniques is developed in their study. Adaptive content management is implemented in a MAR system, where the behaviours function in a modular system can combine and match individual user behaviours to the visual outputs in AR, and a final minimal set of useful behaviours is proposed that can be easily controlled by the user in a variety of mobile and stationary tasks [92, 109].

## 3.2 Collaborative UIs in Multi-user and Multi-device AR

Adaptive AR UIs can serve as ubiquitous displays of virtual objects that can be shown anywhere in our physical surroundings. That is, virtual objects can be floating in the air on any physical background, which can be reached out to or manipulated by multiple users with their egocentric views [127]. Users engaged in their AR-mediated physical surroundings are encouraged to accomplish tasks in co-facilitated environments with shared and collaborative AR experiences amongst multiple users [147]. Multiple dimensions of AR collaborative UIs are discussed throughout various applications, for instance, working [147, 236] and playful [62, 127] contents, local/co-located [180] and remote [210] users, sole AR [147, 180] and a mixture of AR and VR [62, 194], co-creation/co-editing by multiple users (i.e., multiple users at the front of AR scenes) [147, 180], supported and guided instruction (i.e., multiple users connecting to one user at the front of AR scenes) [52, 62], human-to-human interactions [127, 147], and the interaction between human and AR bots, such as tangible robots [186] and digital agent representatives [17].

Multi-user collaborative environments have been a research topic in human-computer interaction, which has evolved from sedentary desktop computers [240] to mobile devices and head-worn computers [147, 236]. The success of such collaborative environments needs to cope with several design challenges, including (1) high awareness of others' actions and intentions; (2) high control over the interface; (3) high availability of background information; (4) information transparency amongst collaborators while preventing user privacy leakage; and the impact of user features on other users [38, 286], e.g., how users perceive or can interact with different users'

Just Accepted

ongoing collaborative experience. These design challenges of awareness, control, availability, transparency, and privacy serve as fundamental issues enabling multiple users to interact with others in collaborative and shared environments smoothly. When the collaborative and shared environments are deployed to AR, features such as enriched reality-based interaction and high levels of adaptability are introduced. As discussed previously, the additional design challenges extend from resolving multi-user experiences to reality-based interaction across various AR devices, including AR/VR headsets, smartwatches, smartphones, tablets, large-screen displays, and projectors. These challenges prioritise exploring the management of multiple devices and their platform restrictions, unifying the device-specific sensing and their interaction modalities, and connecting the collaborative AR environments with physical coordinate systems in shared views [236]. The complexity of managing various devices leads to the need for an AR framework to systematically and automatically enable user collaborations in co-aligned AR UIs. It is important to note that the majority of evaluation frameworks focus on small numbers of quantitative metrics, such as completion time and error rate in an example of visual communication cues between an on-site operator and a remote-supporting expert [125], and they neglect the multi-user responses to the physical environments.

### 3.3 Evaluation Metrics for AR UIs

AR interfaces were initially considered for industrial applications, and the goodness or effectiveness of such augmentations was limited to work-oriented metrics [56]. A very early example refers to augmented information, such as working instructions for front-line labourers on assembly lines, where productivity, work quality, and work consistency are regarded as evaluation metrics [71]. However, these work-orientated metrics are not equivalent to user experience (e.g., the easiness of handling the augmentation) and neglect the critical aspects, especially user-centric metrics. The multitudinous user-centric metrics are generally inherited from traditional UX design issues, which can be categorised into four elements: (1) user perception of information (e.g., whether the information is comprehensible, understandable, or can be easily learnt), (2) manipulability (i.e., usability, operability), (3) task-oriented outcome (e.g., efficiency, effectiveness, task success), and (4) other subjective metrics (e.g., attractiveness, engagement, satisfaction in use, social presence, user control) [19]. When augmentations are displayed on handheld devices, such as smartphones and tablets, ergonomic issues, such as ease in AR content manipulation with two-handed and one-handed operations, are further considered [222]. Nowadays, AR is deployed in real-world scenarios, primarily acting as marketing tools, and hence business-orientated metrics are further examined, such as utility, aesthetic, enjoyment, and brand perception [207, 257]. Additionally, multi-user collaborative environments require remote connections in AR [291]. Lately, the quality of experience (QoE) through computation offloading mechanisms to cloud or edge servers encounter new design challenges of AR UIs [221, 251].

The user perception of AR information which leads to the comprehensibility of AR cues, and the learnability of AR operations in reality-based interactions has been further investigated as a problem of multi-modal cues, such as audio, video, and haptics in various enriched situations driven by AR [77]. The intelligent selection of AR information and adaptive management of information display and multi-modal cues are crucial to users' perception of AR environments [47, 133]. This can be considered a fundamental issue of interface plasticity. In the mixed contents between digital and physical realities [19], the plasticity of AR interfaces refers to the compatibility of the information to physical surroundings as well as situations of users (i.e., context-awareness) [78, 143].

After defining evaluation metrics, AR UI practitioners (e.g., software engineers and designers) often examine the dynamic user experience in interactive environments while satisfying the metrics mentioned earlier. There are attempts to assess the AR experience by building mini-size studio interactive spaces to emulate AR environments [124]. However, such physical setups and iterative assessments are costly and time-consuming, primarily when AR is implemented on large scales (i.e., ubiquitously in our living spaces). Moreover, the increasing number

Just Accepted

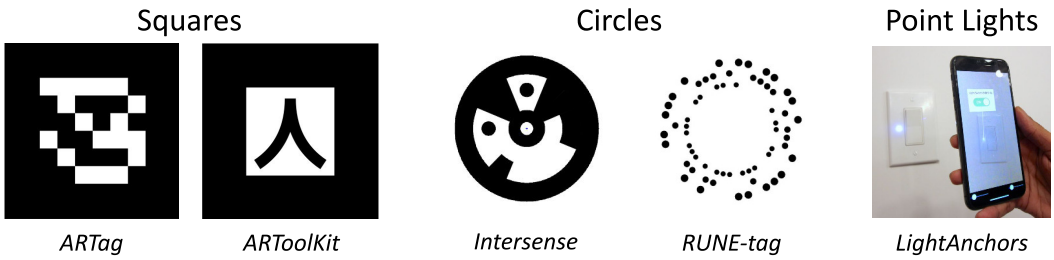


Fig. 2. Examples of square, circular, and point light fiducial markers used in MAR.

of evaluation metrics calls for systematic evaluations of AR UIs [107]. It is, therefore, preferable to assess AR UI design metrics through systematic approaches and even automation, with prominent features of real-time monitoring of system performance, direct information collection via user-device interaction in AR, and more proactive responses to improve user perceptions of AR information [107]. However, to our knowledge, the number of existing evaluation frameworks is minimal, and their scopes are limited to specific contexts and scenarios, such as disaster management [175]. More generic evaluation frameworks with high selectivity of AR evaluation metrics pose research opportunities in the domain of AR interface designs [77, 107].

## 4 MAR FRAMEWORKS

MAR systems can be deployed individually on client devices (e.g., smartphones and HMDs) without the need for separate hardware or client devices and external servers. This section investigates existing MAR frameworks, many of which are designed for creating MAR applications. A general hardware benchmark for these frameworks can be found by exploring the minimum requirements for MAR frameworks, such as ARCore and ARKit. These software development kits (SDKs) should be optimised for the devices released by Google and Apple. According to the device specification for both SDKs, the oldest devices still actively supported were released in 2015, i.e., the Nexus 5X [83] and the iPhone 6S [15]. However, older devices can still be supported by commercial and open-source solutions, such as ARToolKit, which supports 2013-released Android devices [55], and Vuforia, which supports the 2012-released iPhone 5 [202]. Through software optimisation, MAR support for various mobile hardware is possible. However, hardware backwards compatibility can be reduced as more software features are developed. Those features may require better processing capabilities or additional hardware, which may not always be available from older devices. Therefore, features may be omitted from version releases or device support removed completely.

### 4.1 Evaluation Metrics for MAR Frameworks

MAR frameworks have various features that aim to support different platforms and constraints. We compare these frameworks by evaluating the availability of the features, capabilities, and devices on which the frameworks can be deployed as evaluation metrics.

(1) **Platform support:** MAR is not necessarily restricted to only smartphones running Android ( ) or iOS ( ) operating systems (OSs). Wearables and HMDs are other MAR devices, which run alternative OSs, such as Linux ( ) [69, 192] and Windows Holographic OS/Windows 10 ( ) [165]. The frameworks are typically deployable on several different hardware types. Cross-platform support is also found through web browser-based MAR ( ) and using the cross-platform Unity Real-Time Development Platform ( ) [126].

Just Accepted

(2) **Tracking:** For providing MAR experiences, virtual annotations are placed and tracked within physical environments. The two most commonly used techniques to enable and produce these experiences are (a) marker- or fiducial-based, and (b) markerless tracking.
- Marker or fiducial tracking uses 2D patterns of predefined shapes and sizes placed within environments. These are recognised from optical images, and accompanying information associated with the patterns is returned to users [66]. Among the different types of planar markers used in AR, Figure 2 presents typical examples, including shaped-based markers such as squares (ARTag [66], ARToolKit [134]), circles (Intersense [170], RUNE-tag [32]), and pervasive point lights (e.g., LEDs or light bulbs) to spatially-anchor data (LightAnchors [5]). Planar patterns, such as QR codes [116] are good at encoding information. However, they do not function well in certain MAR situations, such as when there are large fields of view and when perspectives are distorted [66].
- Markerless tracking utilises local environments of users to generate and fix virtual markers for 3D tracking [65, 176]. Also known as natural feature tracking (NFT), the natural features are points and regions of interest which are detected and selected, then motion and pose estimation are performed to track those features and augmented content can be fixed to physical environments [176]. The most popular technique for this environment analysis is simultaneous localisation and mapping (SLAM) [215]. There are several SLAM methods, but they all generally reconstruct an environment into a 3D spatial map (3D point cloud) to build a global reference map while simultaneously tracking the subject's position [27, 48]. For example, visual SLAM (vSLAM) methods typically use cameras to collect input data, camera motion estimation is then performed, along with 3D structure estimation, global optimisation, and relocalisation for mapping and tracking [244]. Markerless tracking has advantages such as not requiring prior knowledge of a user's locale and not needing to place additional objects (e.g., fiducial markers) in an environment to create MAR experiences.

Some frameworks also support a variety of other tracking techniques and behaviours, which have varying degrees of support across the different frameworks [254], including (a) Device tracking: the device's position and orientation are tracked with respect to physical space. (b) Plane tracking: flat, horizontal and vertical surfaces are detected and tracked. (c) Hand tracking: human hands are recognised and tracked. (d) Body tracking: the entire human body is recognised in relation to physical space. (e) Facial tracking: the human face is detected and tracked.

(3) **Features:** Frameworks and SDKs support various features, which enhance the overall MAR experience for users, whether through the quality of service or quality of experience improvements [254]: (a) Point clouds: feature point maps from NFT, frameworks sometimes allow the maps to be accessible to developers and users. (b) Anchors: these are points and locations of arbitrary position and orientation the system tracks. (c) Light estimation: the average colour temperature and brightness within a physical space is estimated. (d) Environment probes: probes which are used to generate a cube map that represents a particular area of a physical environment. (e) Meshing: a physical space is virtually segmented and masked using triangular meshes. (f) Collaboration: positions and orientations of other devices are tracked and shared to create a collaborative MAR experience. (g) Occlusion: objects in the physical world have known or calculated distances to aid in the rendering of 3D content, which allows for realistic blending of physical and virtual worlds. (h) Raycasting: the physical surroundings are queried for detected planes and feature points. (i) Pass-through video: the captured camera feed is rendered onto the device touchscreen as the background for MAR content. (j) Session management: the platform-level configuration is changed automatically when AR features are enabled or disabled.

(4) **Sensors:** The hardware providing MAR experiences are now equipped with a wide range of sensors. In addition, they contain high-resolution displays and cameras; inertial measurement unit (IMU) sensors such as gyroscopes, accelerometers, and compasses; and Global Positioning System (GPS) sensors are now

Just Accepted

a standard inclusion in smartphones and HMDs. They benefit MAR frameworks by estimating the user location and orientation to display location-relevant information [45]. Ranging technologies, such as LiDAR (light detection and ranging) sensors, are now used to enhance current arrays of optical cameras on mobile smartphones for better spatial mapping, which is available on recent Apple iPhone and iPad devices [104].

(5) **Architecture:** With every iteration release of mobile hardware, these devices are more capable with increasingly powerful CPUs and GPUs to perform onboard tasks (*offline* processing). However, not every user can access the latest and most powerful devices. Therefore in these scenarios, a framework could employ external servers to offload computations (*online* processing). Frameworks could then support either offline or online recognition of objects or both.

Other considerations for selecting a MAR framework include whether the framework is available for free or requires the purchasing of a commercial software license; whether the framework is open source for developers to expand and build upon; whether the framework is for general "all-purpose" usage, i.e., not just for one use-case like facial-tracking; and whether there are studio tools to simplify and aid in the development of MAR experiences.

## 4.2 Frameworks

We provide a comprehensive summary of existing MAR frameworks and SDKs, considering the evaluation metrics specified in Section 4.1. We tabulate our results in Table 1. To provide a basic filter on the frameworks and SDKs listed, we exclude frameworks that have not been updated within four years. This represents the age of the iPhone SE (1st generation, 2016) which is supported by the latest version of iOS 14 from Apple (i.e., the oldest supported iPhone) [**?** ]. Similarly, Google supports the Pixel 2 (2017) in their latest Android 11 OS update [84]. Frameworks that last had updates four years ago may not function on the wide range of new hardware and software now available to users. Certainly, this may also be true for frameworks that have not had updates since one year ago, these are therefore marked in Table 1 with an asterisk (*). Readers of this survey can find additional information on these listed frameworks in the online appendix (Appendix A), and they can find a continually updated version of the table at https://jackycao.co.uk/research/ar_survey/.

From the 37 listed frameworks, Android and iOS are the two most consistently supported platforms. Support for Unity is also often available as it simplifies the MAR application development process, e.g., easily exporting produced applications to multiple platforms. In terms of support for different tracking technologies: (1) 30 of the 37 frameworks support 2D image marker tracking, (2) 27 supports natural feature tracking, (3) only 6 support hand tracking, (4) facial tracking is found in 8 of the frameworks, and (5) only 2 frameworks support 2D and 3D body tracking. This indicates that there is still a primary focus on environment-based MAR, and human body-based tracking is still of limited interest to framework developers.

By comparison, there is varied support for the different features in frameworks. Both ARCore and ARKit support most of the described features, forming the baselines upon which other frameworks can be compared and built. Among the ten listed features, the following are organised as the most-to-least supported: (1) pass-through video, (2) session management, (3) anchors, (4) point clouds and occlusion, (5) raycasting, (6) meshing, (7) light estimation, and (8) collaboration and environmental probing. While the overall average number of supported features is approximately 3 per framework, ten frameworks only support 1, i.e., pass-through video. Concerning the type of architecture in use, offline and on-device-based processing is the most popular with 21 frameworks, 9 frameworks use purely an online-based processing architecture, and 7 frameworks can be both offline- or online-based. The online frameworks are often obtainable as paid services which are offered alongside a free trial variant, while the offline frameworks are generally free, albeit not always open source.

Just Accepted

Table 1. Comparisons of several available features in MAR frameworks and SDKs.

| *Framework/SDK* | *Platform support* | | | | | | *Tracking* | | | | | | | *Features* | | | | | | | | | | | *Sensors* | | | | *Others* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Markers | NFT | Device | Plane | Hand | 2D & 3D body | Facial | Point clouds | Anchors | Light estimation | Environment probes | Meshing | Collaboration | Occlusion | Raycasting | Pass-through video | Session management | Camera | LiDAR | IMU | GPS | Architecture | Price | Open source | General | Studio tool |
| A-Frame (1.2.0) [1] | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | Online | Free | ✓ | ✓ | ✗ |
| ALVAR (0.7.2) [265] | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | Offline | Free | ✗ | ✓ | ✗ |
| Amazon Sumerian (N/A) [10] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Online | Free, Paid | ✗ | ✓ | ✓ |
| ApertusVR (0.9*) [243] | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| ARCore (1.23.0) [82] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Offline | Free | ✓ | ✓ | ✗ |
| ARKit (4) [16] | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Offline | Free | ✗ | ✓ | ✓ |
| ARMedia SDK (2.1.0*) [105] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | Offline | Paid | ✗ | ✓ | ✗ |
| ARToolKit (5.4*) [120, 134] | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| artoolkitX (1.0.6.1) [212] | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| ArUco (3.1.12) [14, 218] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| AR.js (3.3.1) [64] | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | Online | Free | ✓ | ✓ | ✗ |
| Augment (4.0.6*) [21] | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | Online | Paid | ✗ | ✗ | ✓ |
| Augmented Pixels (N/A) [22] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | Offline | Paid | ✗ | ✓ | ✗ |
| AugmentedPro (2.4.3) [23] | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Paid | ✗ | ✓ | ✓ |
| Banuba (0.35.0) [29] | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | Offline | Paid | ✗ | ✗ | ✗ |
| Blippar (N/A) [35] | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Online | Paid | ✗ | ✓ | ✓ |
| CraftAR (5.2.1*) [44] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | Both | Free, Paid | ✗ | ✓ | ✓ |
| DeepAR (2.3.1) [102] | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Free, Paid | ✗ | ✗ | ✓ |
| EasyAR (4.2.0) [261] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | Offline | Free, Paid | ✗ | ✓ | ✗ |
| HERE SDK (3.17) [94] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Offline | Paid | ✗ | ✗ | ✗ |
| Kudan AR SDK (1.6.0) [283] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | Offline | Free, Paid | ✗ | ✓ | ✗ |
| Lumin SDK (0.25.0) [152] | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | Offline | Free | ✗ | ✓ | ✗ |
| MAXST AR SDK (5.0.3) [160] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | Both | Free, Paid | ✗ | ✓ | ✗ |
| Minsar (2.0) [182] | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Both | Free, Paid | ✗ | ✓ | ✓ |
| MRTK (2.6.0) [166] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| NyARToolkit (5.0.8*) [203] | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | Offline | Free | ✓ | ✓ | ✗ |
| Onirix (N/A) [181] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Online | Paid | ✗ | ✓ | ✓ |
| Pikkart AR SDK (3.5.8*) [193] | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Both | Free, Paid | ✗ | ✓ | ✗ |
| PlugXR (1.0.0) [198] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | Online | Paid | ✗ | ✓ | ✓ |
| Universal AR SDK (N/A) [287] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Online | Free | ✗ | ✓ | ✗ |
| Vectary (N/A) [258] | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Online | Paid | ✗ | ✗ | ✓ |
| Vidinoti SDK (N/A) [259] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | Both | Paid | ✗ | ✓ | ✓ |
| ViewAR SDK (N/A) [260] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Offline | Paid | ✗ | ✓ | ✗ |
| Vuforia (9.7) [201] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | Both | Free, Paid | ✗ | ✓ | ✗ |
| Wikitude (9.6) [275] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | Both | Paid | ✗ | ✓ | ✓ |
| WebXR (N/A) [30] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ❍ | ❍ | ❍ | ❍ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ❍ | ❍ | ❍ | Offline | Free | ✓ | ✓ | ✗ |
| XZIMG (2.0.2*) [284] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | Offline | Free, Paid | ✗ | ✓ | ✗ |

Frameworks and SDKs marked with an asterisk (*) have not been updated in 1+ years.
Features or functions optionally supported in different platforms are marked with ❍.

Just Accepted

## 5 MACHINE LEARNING METHODS FOR MAR

Different MAR frameworks and implementations contain approximately the same basic components, irrespective of the hardware used or whether the system is self-contained on one device or distributed between the client and an external server. Figure 1 in Section 1 presents the typical pipeline of MAR systems where different tasks function together to create MAR experiences. Several of these tasks often require ML algorithms to be successfully run, namely, machine vision-based tasks. ML is a set of methods and techniques used for modelling or building

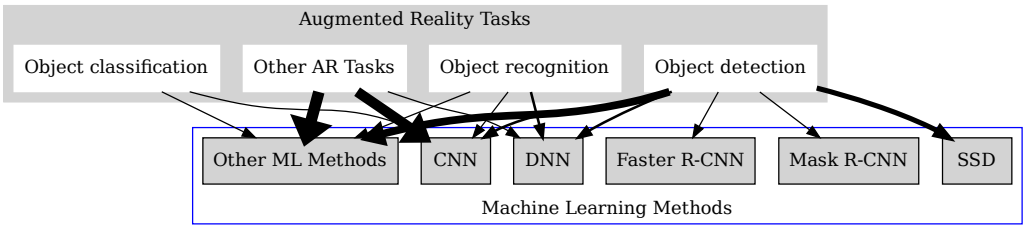


Fig. 3. Mapping of AR tasks found in the works of Table 2 to the ML methods used to fulfil the tasks. The width thickness of the arrow lines denotes the number of works associated with the tasks-to-ML method.

programs that employ knowledge gathered from solving previous problems to answer unseen examples of similar problems, i.e., learning from past experiences [164, 167]. ML technologies are crucial for MAR adoption because they make the processing of visual, audio, and other sensor data more intelligent while simultaneously protecting privacy and security, for example, using ML methods to perform distributed model training and not requiring privacy or security-sensitive data to be sent from the device (i.e., as with federated learning). A significant subfield of ML is deep learning, where multiple layers represent data abstractions to build computational models [199]. Deep learning methods are useful in different domains [3].

In this section, we explore several MAR tasks which require the usage of ML methods. More specifically, we provide the requirements of the tasks, the ML methods used to fulfil the tasks, and the advantages and disadvantages of those methods. Table 2 provides a collection of the most significant works for the AR as mentioned earlier tasks, as well as the ML methods used to fulfil them based on the deployment location of the ML algorithms, i.e., Server, Edge, or Client. Subsequently, Figure 3 provides a summary mapping of these tabulated works, linking the AR tasks to the ML methods. This figure demonstrates the popularity of specific methods, such as CNN and SSD (more details in Section 5.1), and their roles in accomplishing AR tasks.

### 5.1 ML for Object Detection and Object Recognition

Several MAR systems include object detection and/or recognition as the primary pipeline task, which uses ML methods during execution. Object detection supports MAR experiences by allowing the system to understand and detect which objects are in a particular scene at a given moment and then return the spatial location and extent of each object instance [144]. For MAR, in particular, object detection is used to provide environmental context-awareness and system knowledge of what additional augmentations should be provided to users and where to place them on UIs.

Deep learning is used to achieve object detection, where feature representations are automatically learned from data. The most commonly used type of deep learning models for object detection and object recognition are Convolutional Neural Networks (CNN) [8], Deep Neural Networks (DNN) [43], Region-based Convolutional Neural Networks (R-CNN) [80], Faster Region-based Convolutional Neural Networks (Faster R-CNN) [290], You Only Look Once (YOLO) [213], Single Shot MultiBox Detectors (SSD) [146], ResNet [90], Neural Architecture Search Net (NASNet) [297], Mask Region-based Convolutional Neural Networks (Mask R-CNN) [89], DenseNet [100], RetinaNet [142], and EfficientNet [245]. These networks can be used to create algorithms that can produce the same detection or recognition results. The major differences between each neural network are related to the structure and approach of the network when processing input data (e.g., camera images for optical object

Just Accepted

Table 2. List of example AR works which use ML methods for completing AR tasks or as part of the system.

| Device | Overview | Task | ML method |
|---|---|---|---|
| Server | Deep learning for AR tracking [7] | Object detection | CNN |
| | Real-time moving object detection for AR [54] | Object detection | Background-foreground non-parametric-based |
| | Deep-learning-based smart task assistance for wearable AR (HoloLens) [185] | Object detection and instance segmentation | Mask R-CNN |
| | Interacting with IoT devices in AR environment[241] | Real-time hand gesture recognition | (2D) CNN |
| | Edge-assisted distributed DNN for mobile WebAR [216] | Object recognition | DNN |
| | Object detection and tracking for face and eyes AR [88] | Object detection, object recognition | History of Oriented Gradients, Haar-like features |
| | AR surgical scene understanding improved with ML [188] | Object identification | Random forest |
| | Dynamic image recognition methods for AR [51] | Feature extraction, object recognition | CNN, XGBoost |
| | AR platform for interactive aerodynamic design and analysis [26] | Object recognition | Manifold learning |
| | AR retail product identification [255] | Object detection | SSD |
| | Improving retail shopping experience with AR [53] | Object recognition | ResNet50 (CNN) |
| | AR instructional system for mechanical assembly [131] | Object detection | Faster R-CNN |
| | Deep learning for AR [132] | Object tracking, light estimation | CNN |
| | AR for radiology [252] | Image segmentation | CNN |
| | AR design personalisation for facial accessory products [101] | Facial tracking | AdaBoost |
| | Low cost AR for automotive industry [205] | Feature extraction, object classification | Linear SVM, CNN |
| | AR training framework for neonatal endotracheal intubation [293] | Assessing task performance | CNN |
| | AR video calling with WebRTC API [110] | Semantic segmentation | CNN |
| | AR gustatory manipulation [171] | Food-to-food translation | GAN |
| Edge | Edge-based inference for ML at Facebook [281] | Image classification | DNN |
| | Supporting vehicle-to-edge for vehicle AR [295] | Object detection | Deep CNN (YOLO) |
| | AR platform for operators in production environments [253] | Object detection | SSD |
| | Spatial AR with single IR camera [87] | 3D pose estimation, image classification | Hough Forests, Random Ferns |
| | Federated learning for low-latency object detection and classification [50] | Object classification modelling | Federated learning |
| | Optimising learning accuracy in MAR systems [91] | Delay and energy modelling | CNN |
| Client | Application to support users in everyday grocery shopping [266] | Object classification | Random forests |
| | Outdoor AR application for geovisualisation [209] | Object detection | SSD |
| | MAR object detection [137] | Object detection | DNN |
| | Reducing energy drain for MAR [13] | Object detection | DNN |
| | HoloLens surgical navigation [263] | Object detection and pose estimation | CNN |
| | AR inspection framework for industry [189] | Object detection | R-CNN |
| | Learning Egyptian hierogplyphs with AR [197] | Object detection | SSD MobileNets |
| | Interest point detection [238] | Object detection | CNN |
| | Edge-assisted distributed DNN for mobile WebAR [216] | Object recognition | DNN |
| | AR navigation for landmark-based navigation [11] | Prediction of speed of movement | Penalised linear regression |
| | Enhancing STEM education with AR [12] | Object detection | MobileNets |
| | Campus navigation with AR [139] | 3D model placement determination | CNN |
| | AR assisted process guidance on HoloLens [214] | Predicting process quality metrics | Decision tree classification |
| | Indoor AR for Industry 4.0 smart factories [239] | Object detection | MobileNets |
| | AR application for science education of nervous systems [99] | Image classification | CNN |

detection/recognition). For example, R-CNN, which tries to find rectangular regions which could contain objects in an image based on the features of each rectangular region extracted by CNN, is one of the most important approaches for objective detection. Fast R-CNN has been proposed by applying the CNN to the whole image to find a rectangular region with fully connected layers to reduce the computation cost caused by R-CNN. Fast R-CNN is proposed by merging R-CNN with a region proposal network (RPN) to share full-image convolutional features. Mask R-CNN is usually used in segmentation and could be regarded as a mixture of Faster R-CNN and fully convolutional networks for object detection. CNNs and the subsequent variants are powerful components in the AR pipeline for producing AR experiences, without which the system would require more time to analyse the environment, which comes at the cost of user QoE. To handle dynamic MAR scenarios, Su *et al.* [237] propose an architecture leveraging continuous object recognition to handle new types of images without requiring retraining ML models from scratch.

Both object detection and object recognition tasks in MAR systems are typically deployed on servers due to the limitation of client devices and their lack of computation and battery resources to sufficiently sustain acceptable MAR experiences. Examples of server-based deployment include the usage of CNNs for object detection to support AR tracking [7], Mask R-CNN for object detection and instance segmentation to support smart task

Just Accepted

assistance for HoloLens-deployed AR [185], and CNN for object recognition in improving retail AR shopping experiences [53]. Additionally, some works explicitly state their usage of edge servers, citing the processing acceleration gained by the object detection and recognition tasks when a GPU is used to execute the functions, for example, using YOLO to accomplish object detection to support vehicle-to-edge AR [295], and SSD for supporting an edge-based AR platform for operators in production environments [253]. Comparatively, several research works deploy object detection and recognition in an enclosed client system, i.e., without needing to offload computation components to an external server or device. CNNs and DNNs are the most commonly attributed ML models among these works. DNN models have been used to fulfil MAR object detection [137] and support the reduction in energy drain for MAR applications [13]. Alternatively, CNNs are used for object detection and pose estimation for HoloLens surgical navigation [263], interest point detection for an AR application [238], as well as for object detection in an industrial AR inspection framework [189], albeit the model for this latter work is R-CNN. The deployment of these models on the client is supported by increasing efforts to produce more efficient neural networks and reduce their implementation size by using ML libraries specifically designed for deployment on mobile devices, i.e., TensorFlow Lite [250].

### 5.2 ML for Object Tracking

Object tracking in the MAR pipeline allows for objects in the physical world to be located and tracked across the MAR system, which subsequently supports the placement and tracking of virtual annotations derived from the analysis of the surrounding user environments in object detection and object recognition routines. As discussed in Section 4, two commonly used object tracking techniques are marker-based and markerless tracking. Similar to object detection and object recognition, object tracking can be achieved through analysis of optical camera data [272]. However, object tracking differs from object detection as objects must be tracked over time while the MAR experience occurs. In contrast, object detection is more concerned about detecting objects in every frame or when they first appear [285].

Some ML-based approaches for object tracking are Deep Regression Networks [93], Recurrent YOLO (ROLO) [178], Multi-Domain Networks (MDNet) [172], Deep SORT [279], SiamMask [268], TrackR-CNN [262], Deep Regression Learning [74], Adversarial Learning [233], Tracktor++ [33], and Joint Detection and Embedding (JDE) [269]. As evident by some of their names, these object tracking methods are based on deep learning models for object detection and recognition. For example, ROLO combines the object detection utility of a YOLO network with a Long Short-Term Memory (LSTM) network for acquiring the trajectory of an object. The combined ROLO network can track spatial and temporal domains while dealing with occlusion and motion blur effects. TrackR-CNN uses Mask R-CNN for initial object detection and then applies 3D convolutions to incorporate temporal context from the input data streams to form the object tracker and detection and tracking. Meanwhile, TrackR-CNN can also perform segmentation within a single convolutional network. These techniques are not typically found in works that contain object tracking for a MAR system. Rather, custom computer vision tracking algorithms combine deep learning-based object detection and recognition models to track objects across MAR scenes.

One such AR system which uses deep learning detection and a non-deep learning tracker is proposed by Ahmadyan *et al.* [4], where they use a CNN to initialise the pose of an object and then a planar tracker using sensor data to track the object in 9 degrees-of-freedom (DoF, i.e., orientation, translation, and physical size). The 3D tracking is performed on the client device but utilises the mobile GPU to enhance the tracking performance and increase the rate the tracker can run. Alternatively, SLAM is a computer vision algorithm often used in MAR to map and track environments. SLAM uses sensor data from mobile devices such as gyroscopes and accelerometers to achieve reliable tracking [191]. Sernani *et al.* [226] use SLAM for a self-contained tourism AR application, where SLAM localisation allows the system to locate the user's orientation and assists in the

Just Accepted

superposition of arrows and icons to guide the user's attention in an environment. On the other hand, Rambach *et al.* [206] investigate the usage of model-based tracking as an alternative to SLAM tracking as extracting real-world scale information and displaying object-specific AR content may not always be possible with SLAM. Model-based tracking is achieved by using a predefined model or 3D reconstructed textured model of the tracking object and then matching it with a live view of the object during tracking to uncover the 6 DoF pose of the object. Similar to previously mentioned MAR systems, they deploy the computation-heavy object recognition and tracking tasks to a high-performance edge server enabled with a GPU to reduce the total system latency. The client is then free to perform solely data capturing and annotation rendering tasks.

Compared with purely deep learning-based object trackers, these latter systems fuse a deep learning object detection routine with a non-deep learning tracking system. The usage of ML-based methods for tracking is less common when compared to object detection and recognition as there is generally a lack of tracking training data suitable for creating deep learning tracking models [129]. However, exploratory works now use recurrent neural networks (RNN) for implicit learning of temporal dependencies while tracking data.

### 5.3 ML for Adaptive UIs

This section describes the ML methods applied to adaptive MAR UIs.

**InfoF**: The limitations of traditional information filtering methods are twofold [249], i.e. (1) the methods can be destructive and cause data loss for data visualisation, and (2) the methods may fail when the amount of data grows. To avoid these limitations, the recent trend is to use ML methods for massive content layouts and clustering. The advantages of ML methods are summarised by Sahu *et al.* [220], firstly, more efficient and effective rendering is gained when using ML-based layout creation, and secondly, using deep learning-based data clustering offers better performance when transforming data into clustering-friendly nonlinear representations. The information filtering, clustering, and rendering stages in MAR could be integrated as a context-aware recommendation system, with the ML methods for generating relevant user recommendations based on their contextual situation. Recent works apply deep learning architectures for collaborative filtering in recommendation systems [36]. Their results show strong improvements in the prediction and recommendation of information for users due to incorporating context-awareness, user customisation, expert knowledge, and flexibility into recommendation systems [61]. Context-awareness modules in AR can utilise the 5W1H (i.e., the questions of What, Why, When, Where, Who, How) context model to determine what kind of contextual information is to be adopted [98]. ML models can parse context information and decide the model deployed in such contexts. It is worthwhile mentioning that the model can represent a set of users' responses and actions in such contexts (i.e., models of logical thinking). Thus, AR recommendation systems with ML models can fit the AR contents based on users' responses and actions. Jacucci *et al.* [108] present a recommendation system for a mixed reality-based urban exploration application. They consider using user profiles and weightings from crowd-sourced ratings and personal preferences. Their ML technique for personalised search and recommendations is based on two main components, (1) a data model that defines the representation of multiple data sources (e.g., content, social, and personal) as a set of overlaid graphs, and (2) a relevance-estimation model that performs random walks with restarts on the graph overlays and computes a relevance score for information items. However, their study does not present any experimental results or performance data. Zhang *et al.* [292] present an aggregated random walk algorithm incorporating personal preferences, location information, and temporal information in a layered graph to improve recommendations in MAR ecosystems. Readme [47] is a real-time MAR recommendation system using an online-based algorithm. They compare their system with the following baselines: (i) k-Nearest Neighbours (kNN), which is based purely on physical distance, and (ii) focus-based recommendations, which infer the user's interests based on the user's viewpoint and attempts to find the most similar objects within the user's current focus. Their evaluation results

Just Accepted

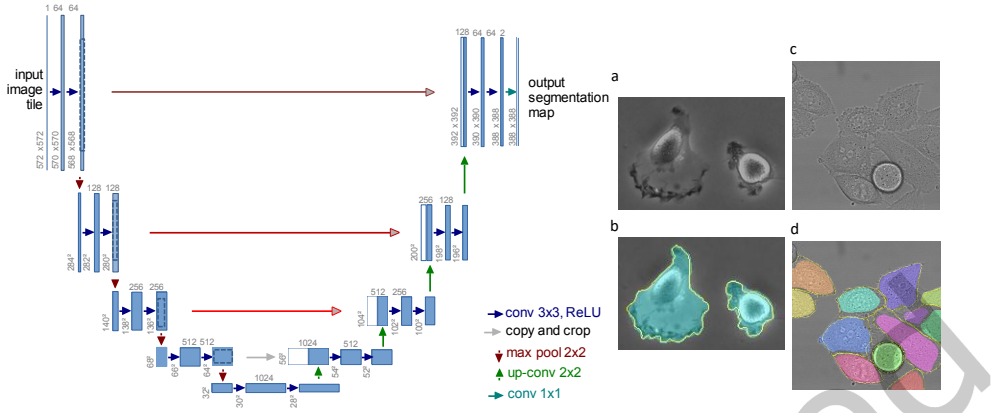


Fig. 4. The left figure presents the architecture of U-Net. The right figures present the result of the ISBI cell tracking challenge. In the right figure, (a) and (c) are input images from the dataset. (b) and (d) are segmentation results using U-Net.

show that ReadMe outperforms these baselines. Alternatively, contextual sequence modelling with RNN is another method to produce recommendation systems in MAR [231].

**RegEA**: Recent works show that applying ML algorithms in safety-critical AR systems could improve tracking accuracy and minimise registration errors. Safety-critical AR systems should provide hints on the user interface to adapt to real-time registration errors to inform users of any potentially harmful or dangerous situations. U-Net [219] is a modified full CNN initially built for precise medical image segmentation. Brunet *et al.* [40] propose a physics-based DNN with U-Net (Figure 4), for providing AR visualisation during hepatic surgery. The physics-based DNN solves the deformed state of the organ by using only sparse partial surface displacement data and achieves similar accuracy as a finite element method solution. They achieve the registration in only 3 ms with a mean target registration error (TRE) of 2.9 mm in the ex vivo liver AR surgery. Their physics-based DNN is 500× faster than a reference finite element method solution. For the RegEA visualisation, this work obtains the surface data of the liver with an RGB-D camera and ground truth data acquired at different stages of deformation using a CT scan. Markers were embedded in the liver to compute TRE. By knowing the TRE values, this work can highlight the region around the 3D model of the virtual liver to alert the user for error minimisation.

**Occlusion Representation and Depth Cues**: Traditional methods for calculating the depth of a scene are ineffective in recovering depth and scene information whenever perceptual cues are used. This is due to the high computational costs of the recovering methods, the under-constrained nature of mobile devices, and the use of limited cues. The advantage of ML methods is their ability to exploit almost all cues simultaneously, thus offering better depth inference, as well as the capability to estimate depth information from RGBE images (i.e., an RGB format that represents a higher dynamic range) in a similar concurrent manner for the completion of other tasks, such as object detection, tracking, and pose estimation [220]. Park *et al.* [185] present a comprehensive smart task assistance system for wearable AR, which adopts Mask R-CNN for several adaptive UI techniques. Their work supports occlusion representation calculation, registration error adaption, and adaptive content placement. Alternatively, Tang *et al.* [247] present GrabAR (Figure 5), a system using a custom compact DNN for generating occlusion masks to support real-time grabbing of virtual objects in AR. The model can calculate and segment the hand to provide visually plausible interactions with objects in the virtual AR environment.

Just Accepted

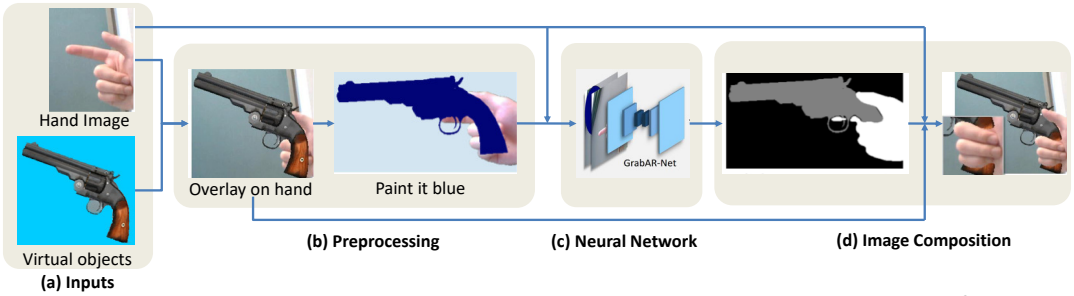


Fig. 5. The workflow of GrabAR for resolving the occlusion between real (hand) and virtual (grabbable) objects in AR views. (a) An image pair of hand and virtual objects are inputs. (b) GrabAR will preprocess them. (c) The preprocessed data is fed into their neural network, where the network predicts an occlusion mask that indicates the hand regions that should overlay above the virtual object. (d) Finally, more natural composition of the real and virtual objects in the AR view is produced.

**IEsti**: The major limitation of classical methods for illumination estimation is the low accuracy problem due to insufficient contextual information, especially the methods using auxiliary information. The methods are computationally expensive and not scalable, which disturbs the real-time performance of AR scene rendering. ML-based illumination estimation resolves the low accuracy and performance issue in AR, which does not require definitive models of the geometric or photometric attributes [220]. DNN approaches are based on the assumption that prior information about lighting can be learned from a large dataset of images with known light sources. Large datasets of panoramas are used to train an illumination predictor, and they use the concept of finding similarities between an input image and one of the projections of individual panoramas [118]. Gardner *et al.* [75] present an illumination estimation solution by calculating diffuse lighting in the form of an omnidirectional image using a neural network. DeepLight [115] is a CNN-based illumination estimation for calculating a dominant light direction from a single RGB-D image. DeepLight uses a CNN to encode a relation between the input image and a dominant light direction. DeepLight applies techniques of outlier removal and temporal filtering to achieve coherence of light source estimation with the consideration of temporal consistency in a time series of video frames on AR devices. DeepLight achieves a higher accuracy for the estimated light direction than the well-known approach proposed by Gardner *et al.* [75], which computes the angular error to the ground-truth light direction and subsequently renders virtual objects in AR using an estimated light source. Moreover, LightNet [174] is the latest illumination estimation using a dense network (DenseNet) architecture, which is trained with two softmax outputs for colour temperature and lighting direction prediction.

## 6 RESEARCH CHALLENGES AND FURTHER DIRECTIONS

We discuss some prominent challenges for MAR, including seamless frameworks for interactable immersive environments, audio AR, user interactions, user interactions with environments, and federated learning. Identifying and analysing these challenges is crucial, and seeking novel theoretical and technical solutions.

### 6.1 Towards seamless frameworks for interactable immersive environments

Existing frameworks and SDKs offer various features to understand physical environments. *ARKit* remarkably achieves a broad coverage of feature recognition, including point clouds, anchors, light estimation, and occlusion. In general, the AR content annotations in existing frameworks already go beyond simple overlays in mid-air. Feature recognition enables these augmented annotations to become a part of the physical world, albeit users can distinguish unnatural positioning and some rendering issues. Interactive AR on human faces, such as with

Just Accepted

Banuba and XZIMG, demonstrates a real-life use case that requires AR overlays to merge with the physical world. We see a noticeable trend for combining AR augmentations in the physical world for different applications, e.g., for medical surgery [40]. One potential research direction could be to extend the different sensor types and employ multi-sensor information to construct user context intelligently.

Alternatively, ML approaches can serve as an automated tool to automatically generate lively and animated digital entities [96] as well as imitate real-world creatures [224, 248]. In addition to this, the collection of ML-generated objects can potentially become a library of animated objects for AR. However, *"how to merge the digital overlaid content (i.e., AR content) interactions with the physical world (e.g., everyday objects and humans)"* are in the nascent stage. The merged AR overlays will eventually co-exist with human users and their surroundings. This leaves us with an unexplored research challenge of designing novel spatial environments. Sample research directions for the next generation of MAR frameworks could be: (1) user issues such as immersive spatial environments and their relationship and interaction among human users, virtual entities, and tangible objects; (2) AI becomes a key actor in spatial environments and how objects (or agents) take actions in an automated fashion; and (3) how to evaluate the interaction of AI-supported animated objects with physical surroundings (i.e., human users and physical objects) without the active participation of designers and software engineers (analogous to the concept of *evaluation probe* [149]).

### 6.2 Audio augmented reality

While most existing frameworks have made significant efforts in visual-based AR cues, audio-based interaction is another neglected and vital aspect. Audio can be regarded as an inescapable design material for audio-based interactions, where individualisation, context-awareness, and diversification are the primary strategies for designing such audio-based interaction [242]. Among three strategies, context awareness (i.e., voices to be tailored to the user's context) aligns with the primary aspect of AR. In addition, audio's sociophonetic aspect should reflect the user's social quality. For example, paying a credit card bill requires audio that gives the user an impression of trustworthiness, honesty, and efficiency [242]. Audio AR will be an interesting topic in examining the alignment of audio and the properties of virtual visuals (e.g., appearance and other contexts) in such audio-driven AR. In brief, audio AR can be presented as either solely audio within the presence of physical spaces or primarily audio with auxiliary cues (i.e., simple visual cues or haptics). Audio AR can initiate spontaneously or be triggered by a user event. Audio can help users reduce their cognitive load on reading visual cues in AR and simultaneously receive information efficiently through audio channels, such as navigation with audio guidance and haptic-based directionality.

### 6.3 User interaction and user workload

Section 4.1 summarises 37 MAR frameworks and SDKs. Generally, these frameworks have not prioritised natural user interfaces (NUIs) [135], and minimal numbers of existing frameworks and SDKs support on-body interfaces such as hand, facial, and body tracking. Although such feature tracking on user bodies enables alternative input modals on AR devices, especially head-worn AR headsets, such discussed frameworks and SDKs do not offer sufficient consideration into the user workload across various types of user interaction tasks, ranging from clicking a button on a 2D menu to manipulating a 3D sphere with rotation movements. More importantly, there exists no integration of user workload measurements into existing frameworks.

As natural user interfaces rely on body movements, user interaction with such interfaces cannot be prolonged due to user fatigue [135, 136]. It is necessary to include computational approaches to understand how the user interacts with AR interfaces to reduce user workload (i.e., collecting user interaction traces). The user interaction trace will be collected for the foundation of data-driven user interaction design, which further offers systematic evaluation modules inside MAR frameworks and SDKs to optimise the user interaction design. The systematic

Just Accepted

evaluation modules should also acquire the ability to assist the iterative design process that allows for natural user interfaces to improve the user workloads in constantly changing AR environments.

### 6.4 Users and their environments

As mentioned above, with user interaction and the potential architecture of collecting data user traces, as these facilities are becoming mature, such frameworks can further consider the user interaction with their tangible environments. This then supports the hyper-personalisation of user experiences, i.e., *letting audiences shape their user interaction designs*. Users with AR devices, e.g., smartphones (as-is) and smartglasses (to-be), encounter digital entities highly customised to the user preference. It is worthwhile mentioning that digital entities will merge with the nearby tangible environments, and user interaction in AR should work in both physical and digital worlds.

The next generation of AR frameworks will encounter two primary challenges with such premises. On the one hand, tangible environments are not static objects. The information and property registration of such entities will not remain static forever, and any changes in such objects will broadly impact the user experience. On the other hand, we cannot guarantee that users will act the same way when interacting with AR augmentations in tangible environments as initially planned by software engineers and product managers when they designed and implemented the AR physical environments, which presumably considered the world to be a static property.

The second challenge is more challenging to address than the first one because the first will eventually be solved by the enhanced sensing capability of our tangible environments and the crowd-based mechanisms that can constantly monitor and reflect the changes in the tangible environments. Regarding the second challenge, we cannot blame such user misalignment between actual user behaviours and the intended design as human error. We have the following example, derived from a classic design example of "*the desired path*", to reinforce the above statement. On a school campus, a user is finding a way to reach a particular building, and two official paths exist (left or right) and a lawn in front of the user. The user with an AR device receives navigational information of either 'going along the path on the left-hand side' or 'going along the path on the right-hand side'. Such navigation information is contextualised with the school campus map and registered in the design and implementation phases. However, due to the incentive of the shortest path, the user eventually takes the diagonal (unofficial) track across the lawn instead of taking any of the official routes. Meanwhile, the AR system keeps reminding the user about the wrong way being taken, according to such pre-defined AR information (referring to the latter challenge). In addition, if such a diagonal path has been repetitively taken, the tangible environment has changed, but the updates of AR information lag behind (referring to the first challenge).

Such uncertainties from dynamic tangible environments and user behaviours give the motivation to build adaptive information management in AR frameworks. Although we have no definitive solutions to address the above challenges at the time being, we identify a research gap in fulfilling the orientation towards user interaction and their environments in adaptive and personalised manners.

### 6.5 Deepening the understanding of users' situations

Nowadays, the frameworks and SDKs provide well-established connections to common sensors such as cameras, inertial measurement units (IMUs), and GPS sensors. This sensing capability allows AR frameworks or systems to detect objects and people within the user's in-situ environments, regardless of marker-based (Figure 2) or markerless approaches. Accordingly, the AR framework makes timely responses and offers user feedback (visual, audio, and haptics), e.g., 3D images popping up on the user's screen. Although research prototypes attempt to manage the user feedback by leveraging the user's geographical information [280], many existing applications are limited by a moderate level of tracking precision due to GPS.

Just Accepted

When AR frameworks are being deployed on a large scale, we foresee the number of candidate AR feedback in one specific location (no less than $5 \times 5$ m$^2$) will surge exponentially. In such a way, AR frameworks may encounter a bottleneck in delivering optimised information flow (i.e., concise and context-aware visuals) on AR devices. In other words, such geographical location-based tracking requires auxiliary channels (e.g., camera-based detection) to get a more precise and detailed understanding of users' environments that guarantee the performance of adaptive user interfaces [133]. In return for improved detection and tracking precision, the framework will take additional overheads. However, the camera-based detection and tracking generate offloading tasks that demand high network traffic, producing significant latency from the distant cloud-based servers. As degraded QoE will significantly impact AR users, we must restructure the client-server architectures in AR frameworks and SDKs.

### 6.6 Privacy preserving MAR frameworks

Existing frameworks have rarely considered multi-user collaboration. One key reason is that such remote collaborations may introduce privacy risks. Sharing user information in practical and just-in-need manners will be a grand challenge. We summarise several challenges for addressing potential privacy leakage in multi-user AR scenarios. Firstly, the next generation MAR frameworks will need a content-sharing module to expose any sensitive information among multiple users across various applications. One remarkable example is sharing user locations in physical space, where multiple users work in the same AR spatial environment. Some users may have their preferred privacy policies, threatening location-based user collaborations. Secondly, the privacy policies of multiple users can conflict with each other. In other words, a user does not realise that one's action could deteriorate the privacy policy of another user in AR, primarily subject to the spatio-temporal relationship. Thirdly, user interaction with AR objects could pose privacy leakage due to its historical record, namely user interaction trace, especially when we consider AR objects as some public properties in these emerging spatial environments. For example, a user may establish some virtual stickers and share them with target users in an AR spatial environment. However, different users may own asynchronous views and hence unparalleled information due to their privacy preferences, and therefore they work differently on the virtual stickers due to different beliefs. Thus, the effectiveness of preventing privacy leakage highly depends on the user-centric and user-interdependent semantics on the application level and a multi-user framework level. It is crucial to strengthen the information-theoretic dimension in AR frameworks.

### 6.7 Federated learning for MAR

The traditional cloud-based MAR frameworks distribute the centralised trained models to AR clients, which may require prohibitively large amounts of time when models are often updated. Federated learning is an emerging ML paradigm to train local models in a decentralised manner and can cooperate with any ML method, which supports personalised models for users and significantly reduces communication latency. Instead of centralising the data to train a global model for improved local predictions, federated learning downloads the global model to capable devices, where models are locally updated. Updates are sent to a central server, aggregating the local updates into an updated global model [50].

For example, to address rigorous network constraints of MAR systems, most existing MAR solutions focus on adaptive viewport schemes, which segment a video into multiple tiles. The system assesses the client's viewport position in the video and predicts the most probable future viewport positions. The server then only transmits the tiles located in the user's field of view and the tiles at the predicted viewport locations. A federated learning approach can train a local viewport prediction model and only transmit the model updates to the server. Such methods perform viewport prediction on-device and request the corresponding tiles from the server. Therefore, a federated learning-based approach allows training of the global model in a distributed fashion while enabling fine-grained personalisation through the personal model.

Just Accepted

## 7 CONCLUSIONS

This survey employs a top-down approach to review MAR frameworks supporting user interfaces and applications. We begin by discussing MAR applications and the content management and generation that supports AR visualisation that is adaptive to the user's environments of high dynamics and mobility. The article then meticulously selects 37 MAR frameworks available in industry or proposed by researchers in academia and further reviews and compares their functions and features. Additionally, the article covers the ML methods in the domain of MAR and provides a trendy analysis of how the methods support MAR in becoming sensitive to user contexts and physical environments. We discussed the significant challenges and unexplored topics of designing seamless yet user-centric MAR frameworks, potentially empowered by ML methods. Finally, this article hopes to provide a broader discussion within the community and invites researchers and practitioners, especially in the fields relevant to MAR, to shape next-generation AR services in the future metaverse era.

## ACKNOWLEDGMENTS

This work is partially funded by Academy of Finland, grant numbers 3196669, 319670, 325774, 326305, and 325570. Xiang Su is the corresponding author.

## REFERENCES

[1] A-Frame. 2020. A-Frame. https://aframe.io/
[2] N. Ab Aziz, K. Ab Aziz, A. Paul, A. Yusof, et al. 2012. Providing augmented reality based education for students with attention deficit hyperactive disorder via cloud computing: Its advantages. In *ICACT*. IEEE, 577–581.
[3] O. Abiodun, A. Jantan, A. Omolara, K. Dada, et al. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018), e00938.
[4] A. Ahmadyan, T. Hou, J. Wei, L. Zhang, et al. 2020. Instant 3D object tracking with applications in augmented reality. *arXiv preprint arXiv:2006.13194* (2020).
[5] K. Ahuja, S. Pareddy, R. Xiao, M. Goel, et al. 2019. Lightanchors: Appropriating point lights for spatially-anchored augmented reality interfaces. In *Proc. UIST*. ACM, 189–196.
[6] A. Ajanki, M. Billinghurst, H. Gamper, T. Järvenpää, et al. 2011. An augmented reality interface to contextual information. *Virtual reality* 15, 2-3 (2011), 161–173.
[7] O. Akgul, H. Penekli, and Y. Genc. 2016. Applying deep learning in augmented reality tracking. In *SITIS*. IEEE, 47–54.
[8] S. Albawi, T. Mohammed, and S. Al-Zawi. 2017. Understanding of a convolutional neural network. In *ICET*. IEEE, 1–6.
[9] Amazon. 2021. Introducing Amazon Salon. https://blog.aboutamazon.co.uk/shopping-and-entertainment/introducing-amazon-salon
[10] Amazon Web Services, Inc. 2020. Amazon Sumerian Overview. https://aws.amazon.com/sumerian/
[11] P. Amirian and A. Basiri. 2016. Landmark-based pedestrian navigation using augmented reality and machine learning. In *Progress in Cartography*. Springer, 451–465.
[12] I. Ang and K. Lim. 2019. Enhancing STEM education using augmented reality and machine learning. In *ICSCC*. IEEE, 1–5.
[13] K. Apicharttrisorn, X. Ran, J. Chen, S. Krishnamurthy, et al. 2019. Frugal following: Power thrifty object detection and tracking for mobile augmented reality. In *Proc. SenSys*. ACM, 96–109.
[14] Aplicaciones de la Visión Artificial. 2020. ArUco. https://github.com/tentone/aruco
[15] Apple Inc. 2020. ARKit. https://developer.apple.com/documentation/arkit
[16] Apple Inc. 2020. Augmented Reality. https://developer.apple.com/augmented-reality/
[17] R. Aramaki and M. Murakami. 2013. Investigating Appropriate Spatial Relationship between User and Ar Character Agent for Communication Using AR WoZ System. In *Proc. ICMI*. ACM, 397–404.
[18] M. Arango, L. Bahler, P. Bates, M. Cochinwala, et al. 1993. The Touring Machine System. *Commun. ACM* 36, 1 (1993), 69–77.
[19] Y. Arifin, T. Galih Sastria, and E. Barlian. 2018. User Experience Metric for Augmented Reality Application: A Review. *Proc. ICCSCI* 135 (2018), 648 – 656.
[20] Arrow Software Development. 2019. Horizon Explorer. https://horizon-explorer.com/
[21] Augment. 2020. 3D and augmented reality product visualization prlatform | Augment. https://www.augment.com/
[22] Augmented Pixels. 2020. Metaverse: Localization, Navigation and 3D Maps Crowdsourcing. http://augmentedpixels.com/
[23] AugmentedPro. 2020. AugmentedPro. https://www.augmentedpro.com/
[24] R. Azuma. 1997. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments* 6, 4 (1997), 355–385.
[25] R. Azuma and C. Furmanski. 2003. Evaluating Label Placement for Augmented Reality View Management. In *Proc. ISMAR*. IEEE, 66.

Just Accepted

[26] A. Badías, S. Curtit, D. González, I. Alfaro, et al. 2019. An augmented reality platform for interactive aerodynamic design and analysis. *Internat. J. Numer. Methods Engrg.* 120, 1 (2019), 125–138.
[27] T. Bailey and H. Durrant-Whyte. 2006. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine* 13, 3 (2006), 108–117.
[28] R. Bakhmut, N. Kunanets, V. Pasichnyk, O. Artemenko, et al. 2021. Using augmented reality WEB-application for providing virtual excursion tours in university campus. In *IEEE CSIT*, Vol. 2. IEEE, 243–247.
[29] Banuba. 2020. Banuba Augmented Reality SDK. https://www.banuba.com/
[30] R. Baruah. 2021. AR and VR Using the WebXR API. (2021).
[31] B. Bell, S. Feiner, and T. Höllerer. 2001. View management for virtual and augmented reality. In *Proc. UIST*. ACM, 101–110.
[32] F. Bergamasco, A. Albarelli, L. Cosmo, E. Rodola, et al. 2016. An accurate and robust artificial marker based on cyclic codes. *IEEE TPAMI* 38, 12 (2016), 2359–2373.
[33] P. Bergmann, T. Meinhardt, and L. Leal-Taixe. 2019. Tracking without bells and whistles. In *Proc. ICCV*. IEEE, 941–951.
[34] S. Bernhardt, S. Nicolau, L. Soler, and C. Doignon. 2017. The status of augmented reality in laparoscopic surgery as of 2016. *Medical image analysis* 37 (2017), 66–90.
[35] Blippar. 2020. Augmented Reality (AR) & Computer Vision Company | Blippar. https://www.blippar.com/
[36] J. Bobadilla, S. Alonso, and A. Hern. 2020. Deep Learning Architecture for Collaborative Filtering Recommender Systems. *Applied Sciences* 10, 7 (2020), 2441.
[37] A. Bornik, R. er, B. Reitinger, G. Gotschuli, et al. 2003. Computer-aided liver surgery planning: an augmented reality approach. In *Medical Imaging 2003*, Vol. 5029. International Society for Optics and Photonics, 395–406.
[38] C. Bowles and M. Wu. 2010. Transparent Collaboration: Letting Users Simulate Another User's World. In *Proc. CHIMIT*. ACM.
[39] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui. 2017. Future Networking Challenges: The Case of Mobile Augmented Reality. In *IEEE ICDS*. IEEE, 1796–1807.
[40] J.-N. Brunet, A. Mendizabal, A. Petit, N. Golse, et al. 2019. Physics-based deep neural network for augmented reality during liver surgery. In *MICCAI*. Springer, 137–145.
[41] F. Bruno, L. Barbieri, E. Marino, M. Muzzupappa, et al. 2019. An augmented reality tool to detect and annotate design variations in an Industry 4.0 approach. *The International Journal of Advanced Manufacturing Technology* 105, 1 (Nov 2019), 875–887.
[42] A. Campos, N. Correia, T. Romão, I. Nunes, et al. 2019. Mobile augmented reality techniques for emergency response. In *Proc. EAI MobiQuitous*. ACM, 31–39.
[43] A. Canziani, A. Paszke, and E. Culurciello. 2016. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678* (2016).
[44] CareAR. 2020. Marker-Based AR Solution. https://carear.com/craftar/
[45] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, et al. 2011. Augmented reality technologies, systems and applications. *Multimedia tools and applications* 51, 1 (2011), 341–377.
[46] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui. 2017. Mobile augmented reality survey: From where we are to where we go. *IEEE Access* 5 (2017), 6917–6950.
[47] D. Chatzopoulos and P. Hui. 2016. Readme: A real-time recommendation system for mobile augmented reality ecosystems. In *Proc. MM*. ACM, 312–316.
[48] C. Chen, W. Chen, J. Peng, B. Cheng, et al. 2017. A real-time markerless augmented reality framework based on SLAM technique. In *ISPAN-FCST-ISCC*. IEEE, 127–132.
[49] D. Chen, C. Tector, A. Br, H. Chen, et al. 1994. Case study: Observing a volume rendered fetus within a pregnant patient. In *Proc. VIS*. IEEE Computer Society Press, 364–368.
[50] D. Chen, L. Xie, B. Kim, L. Wang, et al. 2020. Federated learning based mobile edge computing for augmented reality applications. In *ICNC*. IEEE, 767–773.
[51] Q. Cheng, S. Zhang, S. Bo, D. Chen, et al. 2020. Augmented Reality Dynamic Image Recognition Technology Based on Deep Learning Algorithm. *IEEE Access* 8 (2020), 137370–137384.
[52] M. Cidota, S. Lukosch, D. Datcu, and H. Lukosch. 2016. Workspace Awareness in Collaborative AR Using HMDs: A User Study Comparing Audio and Visual Notifications. In *Proc. AHs*. ACM.
[53] E. Cruz, S. Orts-Escolano, F. Gomez-Donoso, C. Rizo, et al. 2019. An augmented reality application for improving shopping experience in large retail stores. *Virtual Reality* 23, 3 (2019), 281–291.
[54] C. Cuevas, D. Berjón, F. Morán, and N. García. 2012. Moving object detection for real-time augmented reality applications in a GPGPU. *IEEE Transactions on Consumer Electronics* 58, 1 (2012), 117–125.
[55] DAQRI. 2017. Read me for ARToolKit for Android. https://github.com/artoolkit/artoolkit5/
[56] M. de Sa, E. Churchill, and K. Isbister. 2011. Mobile Augmented Reality: Design Issues and Opportunities. In *Proc. MobileHCI*. ACM, 749–752.

Just Accepted

[57] P. Debevec. 2008. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. In *SIGGRAPH*. ACM.
[58] S. Drouin, D. Collins, and M. Kersten-Oertel. 2018. 7 Interaction in Augmented Reality Image-Guided Surgery. In *Mixed and Augmented Reality in Medicine*. CRC Press, 99–114.
[59] R. Du, E. Turner, M. Dzitsiuk, L. Prasso, et al. 2020. DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality. In *Proc. UIST*. ACM, 829–843.
[60] D. Dwyer, C. Ringstaff, and J. S. 1991. Changes in teachers' beliefs and practices in technology-rich classrooms. *Educational leadership* 48, 8 (1991), 45–52.
[61] T. Ebesu and Y. Fang. 2017. Neural citation network for context-aware citation recommendation. In *Proc. SIGIR*. ACM, 1093–1096.
[62] C. Elvezio, M. Sukan, O. Oda, S. Feiner, et al. 2017. Remote Collaboration in AR and VR Using Virtual Replicas. In *SIGGRAPH*. ACM.
[63] S. Erol, A. Jäger, P. Hold, K. Ott, et al. 2016. Tangible industry 4.0: A scenario-based approach to learning for the future of production. *Procedia CIRP* 54 (2016), 13–18.
[64] J. Etienne. 2020. AR.js. https://github.com/jeromeetienne/AR.js/blob/master/README.md
[65] V. Ferrari, T. Tuytelaars, and L. Van Gool. 2001. Markerless augmented reality with a real-time affine region tracker. In *Proc. ISAR*. IEEE, 87–96.
[66] M. Fiala. 2005. ARTag, a fiducial marker system using digital techniques. In *CVPR*, Vol. 2. IEEE, 590–596.
[67] Fondazione Sistema Toscana. 2022. Tuscany+: Augmented Reality Tourism App. http://www.tourismintuscany.it/smartphone-apps/
[68] P. Fraga-Lamas, T. M. Fernández-Caramés, Ó. Blanco-Novoa, and M. A. Vilar-Montesinos. 2018. A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard. *IEEE Access* 6 (2018), 13358–13375.
[69] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, et al. 2011. TranslatAR: A mobile augmented reality translator. In *WACV*. IEEE, IEEE, 497–502.
[70] H. Fuchs, M. Livingston, R. Raskar, K. Keller, et al. 1998. Augmented reality visualization for laparoscopic surgery. In *MICCAI*. Springer, 934–943.
[71] M. Funk, T. Kosch, S. Greenwald, and A. Schmidt. 2015. A Benchmark for Interactive Augmented Reality Instructions for Assembly Tasks. In *Proc. MUM*. ACM, 253–257.
[72] G. Fusco, S. A. Cheraghi, L. Neat, and J. M. Coughlan. 2020. An Indoor Navigation App Using Computer Vision and Sign Recognition. In *Computers Helping People with Special Needs (Lecture Notes in Computer Science)*, K. Miesenberger, R. Manduchi, M. Covarrubias Rodriguez, and P. Peňáz (Eds.). Springer International Publishing, 485–494.
[73] G. Kim. 2011. Augmented Reality for Basel. https://www.perey.com/AugmentedRealityForBasel/
[74] J. Gao, T. Zhang, X. Yang, and C. Xu. 2018. P2t: Part-to-target tracking via deep regression learning. *IEEE TIP* 27, 6 (2018), 3074–3086.
[75] M. Gardner, K. Sunkavalli, E. Yumer, X. Shen, et al. 2017. Learning to predict indoor illumination from a single image. *ACM TOG* 36, 6 (2017), 1–14.
[76] C. Gartenberg. 2021. TikTok debuts first augmented reality filter that uses iPhone 12 Pro's LIDAR camera. https://www.theverge.com/2021/1/7/22217465/tiktok-augmented-reality-filter-iphone-12-pros-lidar-camera
[77] Y. Ghazwani and S. Smith. 2020. Interaction in Augmented Reality: Challenges to Enhance User Experience. In *Proc. ICVARS*. ACM, 39–44.
[78] N. Ghouaiel, J. Cieutat, and J. Jessel. 2014. Adaptive Augmented Reality: Plasticity of Augmentations. In *Proc. VRIC*. ACM.
[79] J. Gibby, S. Swenson, S. Cvetko, R. Rao, et al. 2019. Head-mounted display augmented reality to guide pedicle screw placement utilizing computed tomography. *IJCARS* 14, 3 (2019), 525–535.
[80] R. Girshick, J. Donahue, T. Darrell, and J. Malik. 2015. Region-based convolutional networks for accurate object detection and segmentation. *IEEE TPAMI* 38, 1 (2015), 142–158.
[81] D. Glasgow. 2021. Redefining what a map can be with new information and AI. https://blog.google/products/maps/redefining-what-map-can-be-new-information-and-ai/
[82] Google Developers. 2020. ARCore. https://developers.google.com/ar
[83] Google Developers. 2020. ARCore supported devices. https://developers.google.com/ar/discover/supported-devices#android_play
[84] Google Developers. 2020. Get Android 11 | Android Developers. https://developer.android.com/about/versions/11/get#on_pixel
[85] J. Grubert, T. Langlotz, S. Zollmann, and H. Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE TVCG* 23, 6 (2017), 1706–1724.
[86] D. Han, T. Jung, and A. Gibson. 2013. Dublin AR: implementing augmented reality in tourism. In *IFITT*. Springer, 511–523.
[87] N. Hashimoto and D. Kobayashi. 2016. Dynamic spatial augmented reality with a single IR camera. In *SIGGRAPH*. ACM, 1–1.
[88] Y. Hbali, M. Sadgal, and A. El Fazziki. 2013. Object detection based on HOG features: Faces and dual-eyes augmented reality. In *WCCIT*. IEEE, 1–4.
[89] K. He, G. Gkioxari, P. Dollár, and R. Girshick. 2017. Mask r-cnn. In *Proc. ICCV*. IEEE, 2961–2969.
[90] K. He, X. Zhang, S. Ren, and J. Sun. 2015. Deep residual learning for image recognition. arXiv 2015. *arXiv preprint arXiv:1512.03385* (2015).

Just Accepted

[91] Y. He, J. Ren, G. Yu, and Y. Cai. 2020. Optimizing the learning accuracy in mobile augmented reality systems with CNN. In *ICC*. IEEE, 1–6.
[92] S. Hegde, J. Maurya, A. Kalkar, and R. Hebbalaguppe. 2020. SmartOverlays: A Visual Saliency Driven Label Placement for Intelligent Human-Computer Interfaces. In *Proc. WACV*. IEEE, 1121–1130.
[93] D. Held, S. Thrun, and S. Savarese. 2016. Learning to track at 100 fps with deep regression networks. In *ECCV*. Springer, 749–765.
[94] HERE. 2020. Build apps with HERE Maps API and SDK Platform Access | HERE Developer. https://developer.here.com/
[95] F. Herpich, R. Guarese, and L. Tarouco. 2017. A comparative analysis of augmented reality frameworks aimed at the development of educational applications. *Creative Education* 8, 09 (2017), 1433.
[96] D. Holden, B. Duong, S. Datta, and D. Nowrouzezahrai. 2019. Subspace Neural Physics: Fast Data-Driven Interactive Simulation. In *Proc. SCA*. ACM, 1–12.
[97] K. Holstein, B. McLaren, and V. Aleven. 2019. Designing for Complementarity: Teacher and Student Needs for Orchestration Support in AI-Enhanced Classrooms. In *AIED*. Springer, 157–171.
[98] D. Hong, H. Schmidtke, and W. Woo. 2007. Linking context modelling and contextual reasoning. In *MRC*. Roskilde University, 37–48.
[99] F. Hoyle, H. Dimakuta, K. Cabonita, V. dela Peña, et al. 2019. Nervo: Augmented Reality Mobile Application for the Science Education of Central and Peripheral Nervous Systems. In *HNICEM*. IEEE, 1–7.
[100] G. Huang, Z. Liu, L. Van Der Maaten, and K. Weinberger. 2017. Densely connected convolutional networks. In *Proc. CVPR*. IEEE, 4700–4708.
[101] S. Huang, Y. Yang, and C. Chu. 2012. Human-centric design personalization of 3D glasses frame in markerless augmented reality. *Advanced Engineering Informatics* 26, 1 (2012), 35–45.
[102] I Love IceCream Ltd. 2020. Augmented Reality SDK. https://www.deepar.ai/augmented-reality-sdk
[103] IKEA. 2022. IKEA mobile apps | Customer services. https://www.ikea.com/gb/en/customer-service/mobile-apps/
[104] Apple Inc. 2020. Apple unveils new iPad Pro with LiDAR Scanner and trackpad support in iPadOS. https://www.apple.com/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/
[105] Inglobe Technologies. 2020. AR-media™ Developer Portal. https://dev.inglobetechnologies.com/features/
[106] M. Iqbal. 2019. Pokémon GO Revenue and Usage Statistics (2019). https://www.businessofapps.com/data/pokemon-go-statistics/
[107] S. Irshad and D. Awang. 2018. A UX Oriented Evaluation Approach for Mobile Augmented Reality Applications. In *Proc. MoMM*. ACM, 108–112.
[108] G. Jacucci, S. Andolina, D. Kalkhofen, D. Schmalstieg, et al. 2018. Combining intelligent recommendation and mixed reality in itineraries for urban exploration. *CreMedia* 2017/2 (2018), 18–23.
[109] J. Jia, S. Elezovikj, H. Fan, S. Yang, et al. 2020. Semantic-aware label placement for augmented reality in street view. *The Visual Computer* (2020), 1–15.
[110] R. Jikadra, H. Shah, and P. Kulkarni. 2019. Video Calling with Augmented Reality Using WebRTC API. In *INDIACom*. IEEE, 363–367.
[111] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum, et al. 2000. Information filtering for mobile augmented reality. In *Proc. ISAR*. IEEE, 3–11.
[112] S. Julier, M. Livingston, I. Swan, and D. Brown. 2003. Adaptive user interfaces in augmented reality. In *STARS*.
[113] A. Kamarainen, S. Metcalf, T. Grotzer, A. Browne, et al. 2013. EcoMOBILE: Integrating augmented reality and probeware with environmental education field trips. *Computers & Education* 68 (2013), 545–556.
[114] P. Kán and H. Kaufmann. 2013. Differential irradiance caching for fast high-quality light transport between virtual and real worlds. In *ISMAR*. IEEE, 133–141.
[115] P. Kán and H. Kaufmann. 2019. Deeplight: light source estimation for augmented reality using deep learning. *The Visual Computer* 35, 6 (2019), 873–883.
[116] T. Kan, C. Teng, and W. Chou. 2009. Applying QR code in augmented reality applications. In *Proc. VRCAI*. ACM, 253–257.
[117] K. Kanno, E. Lamounier, A. Cardoso, E. Lopes, et al. 2018. Augmented Reality System for Aiding Mild Alzheimer Patients and Caregivers. In *VR&3DUI*. IEEE, 593–594.
[118] K. Karsch, K. Sunkavalli, S. Hadap, N. Carr, et al. 2014. Automatic scene inference for 3d object compositing. *ACM TOG* 33, 3 (2014), 1–15.
[119] M. Kasper, N. Keivan, G. Sibley, and C. Heckman. 2017. Light source estimation with analytical path-tracing. *arXiv preprint arXiv:1701.04101* (2017).
[120] H. Kato and M. Billinghurst. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. IWAR*. IEEE, 85–94.
[121] H. Kaufmann and D. Schmalstieg. 2002. Mathematics and geometry education with collaborative augmented reality. In *SIGGRAPH*. ACM, 37–41.
[122] M. Kersten-Oertel, P. Jannin, and D. Collins. 2013. The state of the art of visualization in mixed reality image guided surgery. *Computerized Medical Imaging and Graphics* 37, 2 (2013), 98–112.

Just Accepted

[123] A. Khan and S. Khusro. 2015. The rise of augmented reality browsers: Trends, challenges and opportunities. *Pakistan Journal of Science* 67, 3 (2015).
[124] H. Kim and T. Nam. 2015. Augmented Miniature Prototyping Toolkit for UX in Interactive Space. In *Proc. CHI EA*. ACM, 2229–2234.
[125] S. Kim, G. Lee, W. Huang, H. Kim, et al. 2019. Evaluating the Combination of Visual Communication Cues for HMD-Based Mixed Reality Remote Collaboration. In *Proc. CHI*. ACM, 1–13.
[126] S. Kim, H. Suk, J. Kang, J. Jung, et al. 2014. Using Unity 3D to facilitate mobile augmented reality game development. In *WF-IoT*. IEEE, 21–26.
[127] K. Kiyokawa, M. Billinghurst, B. Campbell, and E. Woods. 2003. An Occlusion-Capable Optical See-through Head Mount Display for Supporting Co-Located Collaboration. In *Proc. ISMAR*. IEEE, 133.
[128] C. Kounavis, A. Kasimati, and E. Zamani. 2012. Enhancing the tourism experience through mobile augmented reality: Challenges and prospects. *IJEBM* 4 (2012), 10.
[129] S. Krebs, B. Duraisamy, and F. Flohr. 2017. A survey on leveraging deep neural networks for object tracking. In *ITSC*. IEEE, 411–418.
[130] W. Lages and D. Bowman. 2019. Walking with adaptive augmented reality workspaces: design and usage patterns. In *IUI*. ACM, 356–366.
[131] Z. Lai, W. Tao, M. Leu, and Z. Yin. 2020. Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing. *Journal of Manufacturing Systems* 55 (2020), 69–81.
[132] J. Lalonde. 2018. Deep learning for augmented reality. In *WIO*. IEEE, 1–3.
[133] K. Lam, L. Lee, T. Braud, and P. Hui. 2019. M2A: A Framework for Visualizing Information from Mobile Web to Mobile Augmented Reality. In *PerCom*. IEEE, 1–10.
[134] Lamb, Philip. 2004. ARToolKit. http://www.hitl.washington.edu/artoolkit/projects/
[135] L. Lee and P. Hui. 2018. Interaction methods for smart glasses: A survey. *IEEE Access* 6 (2018), 28712–28732.
[136] L. Lee, K. Lam, Y. Yau, T. Braud, et al. 2019. HIBEY: Hide the Keyboard in Augmented Reality. In *PerCom*. IEEE.
[137] X. Li, Y. Tian, F. Zhang, S. Quan, et al. 2020. Object Detection in the Context of Mobile Augmented Reality. In *ISMAR*. IEEE, 156–163.
[138] H. Liao, T. Inomata, I. Sakuma, and T. Dohi. 2010. 3-D augmented reality for MRI-guided surgery using integral videography autostereoscopic image overlay. *IEEE TBME* 57, 6 (2010), 1476–1486.
[139] C. Lin, Y. Chung, B. Chou, H. Chen, et al. 2018. A novel campus navigation APP with augmented reality and deep learning. In *ICASI*. IEEE, 1075–1077.
[140] C.-H. Lin, Y. Chung, B.-Y. Chou, H.-Y. Chen, et al. 2018. A novel campus navigation APP with augmented reality and deep learning. In *IEEE ICASI*. IEEE, 1075–1077.
[141] T. Lin, H. Duh, N. Li, H. Wang, et al. 2013. An investigation of learners' collaborative knowledge construction performances and behavior patterns in an augmented reality simulation system. *Computers & Education* 68 (2013), 314–321.
[142] T. Lin, P. Goyal, R. Girshick, K. He, et al. 2017. Focal loss for dense object detection. In *Proc. ICCV*. IEEE, 2980–2988.
[143] D. Lindlbauer, A. Feit, and O. Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proc. UIST*. ACM, 147–160.
[144] L. Liu, W. Ouyang, X. Wang, P. Fieguth, et al. 2020. Deep learning for generic object detection: A survey. *IJCV* 128, 2 (2020), 261–318.
[145] Shanhong Liu. 2019. Augmented reality (AR) market size worldwide in 2017, 2018 and 2025 (in billion U.S. dollars). https://www.statista.com/statistics/897587/world-augmented-reality-market-value/
[146] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, et al. 2016. Ssd: Single shot multibox detector. In *ECCV*. Springer, 21–37.
[147] Y. Liu, Y. Zhang, S. Zuo, and W. Fu. 2018. BoatAR: A Multi-User Augmented-Reality Platform for Boat. In *Proc. VRST*. ACM, 1–2.
[148] S. Lu and Y. Liu. 2015. Integrating augmented reality technology to enhance children's learning in marine education. *Environmental Education Research* 21, 4 (2015), 525–541.
[149] A. Luusua, J. Ylipulli, M. Jurmu, H. Pihlajaniemi, et al. 2015. Evaluation Probes. In *Proc. CHI*. ACM, 85–94.
[150] B. MacIntyre and E. Coelho. 2000. Adapting to dynamic registration errors using level of error (LOE) filtering. In *Proc. ISAR*. IEEE, 85–88.
[151] B. MacIntyre, E. Coelho, and S. Julier. 2002. Estimating and adapting to registration errors in augmented reality systems. In *Proc. VR*. IEEE, 73–80.
[152] Magic Leap, Inc. 2020. Developer Portal | Magic Leap. https://developer.magicleap.com/en-us/learn/guides/developer-portal
[153] F. Mahmood, E. Mahmood, R. Dorfman, J. Mitchell, et al. 2018. Augmented reality and ultrasound education: initial experience. *Journal of cardiothoracic and vascular anesthesia* 32, 3 (2018), 1363–1367.
[154] S. Makris, P. Karagiannis, S. Koukas, and A.-S. Matthaiakis. 2016. Augmented reality system for operator support in human–robot collaborative assembly. *CIRP Annals* 65, 1 (2016), 61–64.
[155] M. Markus, F. Fiedrich, J. Leebmann, C. Schweier, et al. 2004. Concept for an integrated disaster management tool. In *Proc. of WCEE*, Vol. 1.
[156] I. Marneanu, M. Ebner, and T. Roessler. 2014. Evaluation of Augmented Reality Frameworks for Android Development. *iJIM* 8, 4 (2014), 37–44.

Just Accepted

[157] J. Martín-Gutiérrez, P. Fabiani, W. Benesova, M. Meneses, et al. 2015. Augmented reality to promote collaborative and autonomous learning in higher education. *Computers in human behavior* 51 (2015), 752–761.
[158] J. Martín-Gutiérrez, J. Saorín, M. Contero, M. Alcañiz, et al. 2010. Design and validation of an augmented book for spatial abilities development in engineering students. *Computers & Graphics* 34, 1 (2010), 77–91.
[159] R. Masoni, F. Ferrise, M. Bordegoni, N. Gattullo, et al. 2017. Supporting remote maintenance in industry 4.0 through augmented reality. *Procedia Manufacturing* 11 (2017), 1296–1302.
[160] MAXST Co., Ltd. 2020. MAXST | The most convenient and the easiest AR software development kit. http://maxst.com/#/en/arsdk
[161] A. McNamara, K. Boyd, J. George, W. Jones, et al. 2019. Information Placement in Virtual Reality. In *VR&3DUI*. IEEE, 1765–1769.
[162] A. McNamara and C. Kabeerdoss. 2016. Mobile augmented reality: Placing labels based on gaze position. In *ISMAR*. IEEE, 36–37.
[163] T. Merel. 2019. For AR/VR 2.0 to live, AR/VR 1.0 must die. https://techcrunch.com/2019/01/14/for-ar-vr-2-0-to-live-ar-vr-1-0-must-die/
[164] D. Michie, D. Spiegelhalter, and C. Taylor. 1994. Machine learning. *Neural and Statistical Classification* 13, 1994 (1994), 1–298.
[165] Microsoft. 2019. Microsoft HoloLens | Mixed Reality Technology for Business. https://www.microsoft.com/en-us/hololens
[166] Microsoft. 2020. microsoft/MixedRealityToolkit-Unity. https://github.com/microsoft/MixedRealityToolkit-Unity
[167] S. Miraftabzadeh, F. Foiadelli, M. Longo, and M. Pasetti. 2019. A survey of machine learning applications for power system analytics. In *EEEIC/I&CPS Europe*. IEEE, 1–5.
[168] D. Mourtzis, V. Zogopoulos, and E. Vlachou. 2017. Augmented reality application to support remote maintenance as a service in the robotics industry. *Procedia CIRP* 63 (2017), 46–51.
[169] D. Mourtzis, V. Zogopoulos, and E. Vlachou. 2018. Augmented reality supported product design towards industry 4.0: a teaching factory paradigm. *Procedia Manufacturing* 23 (2018), 207–212.
[170] L. Naimark and E. Foxlin. 2002. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *ISMAR*. IEEE, 27–36.
[171] K. Nakano, D. Horita, N. Sakata, K. Kiyokawa, et al. 2019. Deeptaste: Augmented reality gustatory manipulation with gan-based real-time food-to-food translation. In *ISMAR*. IEEE, 212–223.
[172] H. Nam and B. Han. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proc. CVPR*. IEEE, 4293–4302.
[173] K. Nantz and T. Lundgren. 1998. Lecturing with technology. *College Teaching* 46, 2 (1998), 53–56.
[174] S. Nathan and M. Beham. 2020. LightNet: deep learning based illumination estimation from virtual images. In *ECCV*. Springer, 568–580.
[175] M. Nebeling, M. Speicher, X. Wang, S. Rajaram, et al. 2020. MRAT: The Mixed Reality Analytics Toolkit. In *Proc. CHI*. ACM, 1–12.
[176] U. Neumann and S. You. 1999. Natural feature tracking for augmented reality. *IEEE TMM* 1, 1 (1999), 53–64.
[177] Niantic. 2021. Pikimin Bloom - Make Every Step Count! https://pikminbloom.com/
[178] G. Ning, Z. Zhang, C. Huang, X. Ren, et al. 2017. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *ISCAS*. IEEE, 1–4.
[179] I. L. Nunes, R. Lucas, M. Simões-Marques, and N. Correia. 2018. Augmented Reality in Support of Disaster Response. In *Advances in Human Factors and Systems Interaction*, Isabel L. Nunes (Ed.). Springer International Publishing, 155–167.
[180] S. Oh, K. Park, S. Kwon, and H. So. 2016. Designing a Multi-User Interactive Simulation Using AR Glasses. In *Proc. TEI*. ACM, 539–544.
[181] Onirix. 2019. SDKs - Onirix. https://www.onirix.com/ar-sdks/
[182] opuscope. 2020. Minsar Studio - Overview. https://www.minsar.app/
[183] E. Ostler. 2013. IPAD/TABLET APPLICATION: SKYORB. *Mathematics and Computer Education* 47, 2 (2013), 154.
[184] G. Papagiannakis, G. Singh, and N. Magnenat-Thalmann. 2008. A survey of mobile and wireless technologies for augmented reality systems. *Computer Animation and Virtual Worlds* 19, 1 (2008), 3–22.
[185] K. Park, S. Choi, M. Kim, and J. Lee. 2020. Deep learning-based mobile augmented reality for task assistance using 3D spatial mapping and snapshot-based RGB-D data. *Computers & Industrial Engineering* 146 (2020), 106585.
[186] Y. Park, Y. Yang, H. Ro, J. Byun, et al. 2018. Meet AR-Bot: Meeting Anywhere, Anytime with Movable Spatial AR Robot. In *Proc. MM*. ACM, 1242–1243.
[187] J. Paulo Lima, R. Roberto, F. Simões, M. Almeida, et al. 2017. Markerless tracking system for augmented reality in the automotive industry. *Expert Systems with Applications* 82 (2017), 100–114.
[188] O. Pauly, B. Diotte, P. Fallavollita, S. Weidert, et al. 2015. Machine learning-based augmented reality for improved surgical scene understanding. *Computerized Medical Imaging and Graphics* 41 (2015), 55–60.
[189] R. Perla, G. Gupta, R. Hebbalaguppe, and E. Hassan. 2016. Inspectar: An augmented reality inspection framework for industry. In *ISMAR*. IEEE, 355–356.
[190] T. Peters, C. Linte, Z. Yaniv, and J. Williams. 2018. *Mixed and augmented reality in medicine*. CRC Press.
[191] J. Piao and S. Kim. 2017. Adaptive monocular visual–inertial SLAM for real-time augmented reality applications in mobile devices. *Sensors* 17, 11 (2017), 2567.
[192] W. Piekarski and B. Thomas. 2001. Augmented reality with wearable computers running linux. In *Australian Linux Conference*. 1–14.
[193] Pikkart. 2019. Pikkart AR SDK & ecosystem. https://developer.pikkart.com/augmented-reality/sdk/

Under Accepted

[194] T. Piumsomboon, Y. Lee, G. Lee, and M. Billinghurst. 2017. CoVAR: A Collaborative Virtual and Augmented Reality System for Remote Collaboration. In *SIGGRAPH*. ACM, 1–2.
[195] C. Plaza de Miguel. 2014. ARLodge: context-aware recommender system based on augmented reality to assist on the accommodation search process. (2014).
[196] ASOS plc. 2020. ASOS scales up use of Augmented Reality in partnership with Zeekit. https://www.asosplc.com/news/asos-scales-use-augmented-reality-partnership-zeekit/
[197] D. Plecher, C. Eichhorn, K. Seyam, and G. Klinker. 2020. ARsinoë-Learning Egyptian Hieroglyphs with Augmented Reality and Machine Learning. In *ISMAR*. IEEE, 326–332.
[198] PlugXR, Inc. 2020. A Cloud based Augmented Reality platform for everyone. https://www.plugxr.com/
[199] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, et al. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM CSUR* 51, 5 (2018), 1–36.
[200] P. Pratt, M. Ives, G. Lawton, J. Simmons, et al. 2018. Through the HoloLens™ looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels. *European radiology experimental* 2, 1 (2018), 2.
[201] PTC. 2020. Equip your workforce with augmented reality to enhance productivity and improve your bottom line. https://www.ptc.com/en/products/augmented-reality
[202] PTC Inc. 2018. Vuforia Fusion Supported Devices. https://library.vuforia.com/articles/Solution/vuforia-fusion-supported-devices.html
[203] R. Iizuka. 2016. Welcome to NyARToolkit.EN | NyARToolkit project. https://nyatla.jp/nyartoolkit/wp/?page_id=198
[204] S. Rajeev, Q. Wan, K. Yau, K. Panetta, et al. 2019. Augmented reality-based vision-aid indoor navigation system in GPS denied environment. In *Mobile Multimedia/Image Processing, Security, and Applications*, Vol. 10993. SPIE, 143–152.
[205] K. Ramasubramaniam and R. Bhat. 2018. LcAR—Low cost augmented reality for the automotive industry. In *ICCE*. IEEE, 1–3.
[206] J. Rambach, A. Pagani, M. Schneider, O. Artemenko, et al. 2018. 6DoF object tracking based on 3D scans for augmented reality remote live support. *Computers* 7, 1 (2018), 6.
[207] D. Rambli and S. Irshad. 2015. UX Design Evaluation of Mobile Augmented Reality Marketing Products and Services for Asia Pacific Region. In *Proc. APCHIUX*. ACM, 42–45.
[208] M. Ranieri. 2020. A new sense of direction with Live View. https://blog.google/products/maps/new-sense-direction-live-view/
[209] J. Rao, Y. Qiao, F. Ren, J. Wang, et al. 2017. A mobile outdoor augmented reality method combining deep learning object detection and spatial relationships for geovisualization. *Sensors* 17, 9 (2017), 1951.
[210] T. Rasmussen and W. Huang. 2019. SceneCam: Using AR to Improve Multi-Camera Remote Collaboration. In *SIGGRAPH*. ACM, 36–37.
[211] P. Rauschnabel, A. Rossmann, and M. C. tom Dieck. 2017. An adoption framework for mobile augmented reality games: The case of Pokémon Go. *Computers in Human Behavior* 76 (2017), 276–286.
[212] Realmax Inc. 2020. artoolkitX. http://www.artoolkitx.org/
[213] J. Redmon and A. Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).
[214] A. Redžepagić, C. Löffler, T. Feigl, and C. Mutschler. 2020. A Sense of Quality for Augmented Reality Assisted Process Guidance. In *ISMAR*. IEEE, 129–134.
[215] G. Reitmayr, T. Langlotz, D. Wagner, A. Mulloni, et al. 2010. Simultaneous localization and mapping for augmented reality. In *ISUVR*. IEEE, 5–8.
[216] P. Ren, X. Qiao, Y. Huang, L. Liu, et al. 2020. Edge-Assisted Distributed DNN Collaborative Computing Approach for Mobile Web Augmented Reality in 5G Networks. *IEEE Network* 34, 2 (2020), 254–261.
[217] C. Robertson and B. MacIntyre. 2004. Adapting to registration error in an intent-based augmentation system. In *Virtual and Augmented Reality Applications in Manufacturing*. Springer, 147–167.
[218] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer. 2018. Speeded up detection of squared fiducial markers. *Image and Vision Computing* 76 (2018), 38 – 47.
[219] O. Ronneberger, P. Fischer, and T. Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 234–241.
[220] C. K. Sahu, C. Young, and R. Rai. 2020. Artificial intelligence (AI) in augmented reality (AR)-assisted manufacturing applications: a review. *International Journal of Production Research* (2020), 1–57.
[221] S. M. Salman, T. A. Sitompul, A. V. Papadopoulos, and T. Nolte. 2020. Fog Computing for Augmented Reality: Trends, Challenges and Opportunities. In *ICFC*. IEEE, 56–63.
[222] M. Santos, T. Taketomi, C. S, J. Polvi, et al. 2014. A Usability Scale for Handheld Augmented Reality. In *Proc. VRST*. ACM, 167–176.
[223] N. Sayed, H. Zayed, and M. Sharawy. 2011. ARSC: Augmented reality student card an augmented reality solution for the education field. *Computers & Education* 56, 4 (2011), 1045–1061.
[224] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, et al. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *CoRR* abs/1506.02438 (2016).
[225] G. W. Scurati, M. Gattullo, M. Fiorentino, F. Ferrise, et al. 2018. Converting maintenance actions into standard symbols for Augmented Reality applications in Industry 4.0. *Computers in Industry* 98 (Jun 2018), 68–79.

Just Accepted

[226] P. Sernani, R. Angeloni, A. Dragoni, R. Quattrini, et al. 2019. Combining image targets and SLAM for AR-based cultural heritage fruition. In *AVR*. Springer, 199–207.
[227] J. Serván, F. Mas, J. L. Menéndez, and J. Ríos. 2012. Using augmented reality in AIRBUS A400M shop floor assembly work instructions. *AIP Conference Proceedings* 1431, 1 (2012), 633–640.
[228] J. Shuhaiber. 2004. Augmented reality in surgery. *Archives of surgery* 139, 2 (2004), 170–174.
[229] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila. 2021. A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1160–1192.
[230] Six to Start & Naomi Alderman. 2021. Zombies, Run! https://zombiesrungame.com/
[231] E. Smirnova and F. Vasile. 2017. Contextual sequence modeling for recommendation with recurrent neural networks. In *Proc. DLRS*. ACM, 2–9.
[232] Snap Inc. 2019. Lens Studio by Snap Inc. https://lensstudio.snapchat.com/lenses/
[233] Y. Song, C. Ma, X. Wu, L. Gong, et al. 2018. Vital: Visual tracking via adversarial learning. In *Proc. CVPR*. IEEE, 8990–8999.
[234] S. R. Sorko and M. Brunnhofer. 2019. Potentials of augmented reality in training. *Procedia Manufacturing* 31 (2019), 85–90.
[235] S. Sotiriou and F. X Bogner. 2008. Visualizing the invisible: augmented reality as an innovative science education scheme. *Advanced Science Letters* 1, 1 (2008), 114–122.
[236] M. Speicher, B. Hall, A. Yu, B. Zhang, et al. 2018. XD-AR: Challenges and Opportunities in Cross-Device Augmented Reality Application Development. *Proc. ACM Hum.-Comput. Interact.* 2, EICS (June 2018), 1–24.
[237] X. Su, A. Jiang, J. Cao, W. Zhang, et al. 2022. Enabling Continuous Object Recognition in Mobile Augmented Reality. In *27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22 Companion)*. ACM, New York, NY, USA, 42–45.
[238] Y. Su, J. Rambach, N. Minaskan, P. Lesur, et al. 2019. Deep multi-state object pose estimation for augmented reality assembly. In *ISMAR*. IEEE, 222–227.
[239] H. Subakti and J. Jiang. 2018. Indoor augmented reality using deep learning for industry 4.0 smart factories. In *COMPSAC*, Vol. 2. IEEE, 63–68.
[240] C. Sun, S. Xia, D. Sun, D. Chen, et al. 2006. Transparent Adaptation of Single-User Applications for Multi-User Real-Time Collaboration. *ACM Transactions on Computer-Human Interactions* 13, 4 (December 2006), 531–582.
[241] Y. Sun, A. Armengol-Urpi, S. N. R. Kantareddy, J. Siegel, et al. 2019. Magichand: Interact with iot devices in augmented reality environment. In *VR&3DUI*. IEEE, 1738–1743.
[242] S. Sutton, P. Foulkes, D. Kirk, and S. Lawson. 2019. Voice as a Design Material: Sociophonetic Inspired Design Strategies in Human-Computer Interaction. In *Proc. CHI*. ACM, 1–14.
[243] SZTAKI. 2020. ApertusVR. http://apertusvr.org/
[244] T. Taketomi, H. Uchiyama, and S. Ikeda. 2017. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications* 9, 1 (Jun 2017), 16.
[245] M. Tan and Q. Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*. PMLR, 6105–6114.
[246] K. Tanaka, Y. Kishino, M. Miyamae, T. Terada, et al. 2008. An information layout method for an optical see-through head mounted display focusing on the viewability. In *ISMAR*. IEEE, 139–142.
[247] X. Tang, X. Hu, C. Fu, and D. Cohen-Or. 2020. GrabAR: Occlusion-aware Grabbing Virtual Objects in AR. In *Proc. UIST*. ACM, 697–708.
[248] Y. Tassa, T. Erez, and E. Todorov. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IROS*. IEEE, 4906–4913.
[249] M. Tatzgern, V. Orso, D. Kalkofen, G. Jacucci, et al. 2016. Adaptive information density for augmented reality displays. In *VR*. IEEE, 83–92.
[250] Google Brain Team. 2021. TensorFlow Lite | ML for Mobile and Edge Devices. https://www.tensorflow.org/lite
[251] K. Toczé, J. Lindqvist, and S. Nadjm-Tehrani. 2019. Performance Study of Mixed Reality for Edge Computing. In *Proc. UCC*. ACM, 285–294.
[252] L. Trestioreanu, P. Glauner, J. Meira, and M. Gindt. 2020. Using Augmented Reality and Machine Learning in Radiology. In *Innovative Technologies for Market Leadership*. Springer, 89–106.
[253] J. Um, J. Popper, and M. Ruskowski. 2018. Modular augmented reality platform for smart operator in production environment. In *ICPS*. IEEE, 720–725.
[254] Unity Technologies. 2020. About AR Foundation | AR Foundation | 4.0.9 . https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/index.html
[255] G. Upadhyay, D. Aggarwal, A. Bansal, and G. Bhola. 2020. Augmented Reality and Machine Learning based Product Identification in Retail using Vuforia and MobileNets. In *ICICT*. IEEE, 479–485.
[256] S. Vaidya, P. Ambad, and S. Bhosle. 2018. Industry 4.0 – a glimpse. *Procedia Manufacturing* 20 (2018), 233–238.
[257] N. Vaidyanathan. 2020. Augmented Reality in Retail-A Case Study: Technology Implications to Utilitarian, Aesthetic and Enjoyment Values. In *Proc. ICVARS*. ACM, 27–34.
[258] Vectary Inc. 2020. Vectary - The Easiest Online 3D Design and 3D Modeling Software. https://www.vectary.com/

Just Accepted

[259] Vidinoti. 2020. V-SDK - Vidinoti. https://www.vidinoti.com/en/sdk-vidinoti/
[260] ViewAR GmbH. 2020. Introduction - Documentation. https://documentation.viewar.com/docs/sdk/introduction
[261] VisionStar Information Technology (Shanghai) Co., Ltd. 2019. EasyAR. https://www.easyar.com/
[262] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, et al. 2019. Mots: Multi-object tracking and segmentation. In *Proc. CVPR*. IEEE, 7942–7951.
[263] M. von Atzigen, F. Liebmann, A. Hoch, D. E. Bauer, et al. 2021. HoloYolo: A proof-of-concept study for marker-less surgical navigation of spinal rod implants with augmented reality and on-device machine learning. *The International Journal of Medical Robotics and Computer Assisted Surgery* 17, 1 (2021), 1–10.
[264] W. Vorraber, J. Gasser, H. Webb, D. Neubacher, et al. 2020. Assessing augmented reality in production: remote-assisted maintenance with HoloLens. *Procedia CIRP* 88 (2020), 139–144.
[265] VTT Technical Research Centre of Finland Ltd. 2019. Augmented Reality / 3D Tracking | ALVAR. http://virtual.vtt.fi/virtual/proj2/multimedia/alvar/index.html
[266] G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, et al. 2015. MANGO-mobile augmented reality with functional eating guidance and food awareness. In *ICIAP*. Springer, 425–432.
[267] J. Wang, Q. Wang, and U. Saeed. 2018. A visual-GPS fusion based outdoor augmented reality method. In *Proc. ACM SIGGRAPH VRCAI (VRCAI '18)*. ACM, 1–4.
[268] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, et al. 2019. Fast online object tracking and segmentation: A unifying approach. In *Proc. CVPR*. IEEE, 1328–1338.
[269] Z. Wang, L. Zheng, Y. Liu, and S. Wang. 2019. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605* 2, 3 (2019), 4.
[270] Wannaby. 2022. Virtual Try On solutions for fashion industry. https://wanna.fashion
[271] S. Wei, G. Ren, and E. O'Neill. 2014. Haptic and audio displays for augmented reality tourism applications. In *HAPTICS*. IEEE, 485–488.
[272] E. Weng, R. Khan, S. Adruce, and O. Bee. 2013. Objects tracking from natural features in mobile augmented reality. *Procedia-Social and Behavioral Sciences* 97 (2013), 753–760.
[273] T. Whelan, R. Salas-Moreno, B. Glocker, A. Davison, et al. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* 35, 14 (2016), 1697–1716.
[274] M. Wieczorek, A. Aichert, O. Kutter, C. Bichlmeier, et al. 2010. GPU-accelerated Rendering for Medical Augmented Reality in Minimally-invasive Procedures. *Bildverarbeitung für die Medizin* 574 (2010), 102–106.
[275] Wikitude GmbH. 2020. Wikitude Cross Platform Augmented Reality SDK. https://www.wikitude.com/products/wikitude-sdk/
[276] J. Wither, C. Coffin, J. Ventura, and T. Hollerer. 2008. Fast annotation and modeling with a single-point laser range finder. In *Proc. ISMAR*. IEEE, 65–68.
[277] J. Wither, S. Diverdi, and T. Hollerer. 2006. Using aerial photographs for improved mobile AR annotation. In *ISMAR*. IEEE, 159–162.
[278] J. Wither and T. Hollerer. 2005. Pictorial depth cues for outdoor augmented reality. In *ISWC*. IEEE, 92–99.
[279] N. Wojke, A. Bewley, and D. Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP*. IEEE, 3645–3649.
[280] C. Woo and D. Lee. 2011. A Development of a Framework for Building Knowledge based Augmented Reality System. *Journal of the Korea society of computer and information* 16, 7 (2011), 49–58.
[281] C. Wu, D. Brooks, K. Chen, D. Chen, et al. 2019. Machine learning at facebook: Understanding inference at the edge. In *HPCA*. IEEE, 331–344.
[282] H. Wu, J, C.-T. Huang, Z.-R. Huang, Y.-B. Chen, et al. 2020. A Rapid Deployment Indoor Positioning Architecture based on Image Recognition. In *IEEE ICIEA*. 784–789.
[283] XLsoft Corporation. 2019. AR (Augmented Reality) for Mobile Applications. https://www.xlsoft.com/en/products/kudan/index.html
[284] XZIMG Limited. 2019. Home - XZIMG. https://www.xzimg.com/Home
[285] A. Yilmaz, O. Javed, and M. Shah. 2006. Object tracking: A survey. *Acm computing surveys (CSUR)* 38, 4 (2006), 13–es.
[286] N. Yuill and Y. Rogers. 2012. Mechanisms for Collaboration: A Design and Evaluation Framework for Multi-User Interfaces. *ACM Transactions on Computer-Human Interactions* 19, 1 (May 2012), 1–25.
[287] Zappar Ltd. 2020. Universal AR SDK. https://zap.works/universal-ar/
[288] T. Zarraonandia, I. Aedo, P. Díaz, and A. Montero. 2013. An augmented lecture feedback system to support learner and teacher communication. *British Journal of Educational Technology* 44, 4 (2013), 616–628.
[289] T. Zarraonandia, P. Díaz, Á. Montero, I. Aedo, et al. 2019. Using a Google Glass-Based Classroom Feedback System to Improve Students to Teacher Communication. *IEEE Access* 7 (2019), 16837–16846.
[290] J. Zhang, W. Xing, M. Xing, and G. Sun. 2018. Terahertz image detection with the improved faster region-based convolutional neural network. *Sensors* 18, 7 (2018), 2327.
[291] L. Zhang, A. Sun, R. Shea, J. Liu, et al. 2019. Rendering Multi-Party Mobile Augmented Reality from Edge. In *Proc. NOSSDAV*. ACM, 67–72.
[292] Z. Zhang, S. Shang, S. Kulkarni, and P. Hui. 2013. Improving augmented reality using recommender systems. In *Proc. RecSys*. ACM, 173–176.

Just Accepted

[293] S. Zhao, X. Xiao, Q. Wang, X. Zhang, et al. 2020. An Intelligent Augmented Reality Training Framework for Neonatal Endotracheal Intubation. In *ISMAR*. IEEE, 672–681.
[294] J. Zhou and K. Yamamoto. 2016. Development of the system to support tourists excursion behavior using augmented reality. *Development* 7, 7 (2016).
[295] P. Zhou, W. Zhang, T. Braud, P. Hui, et al. 2019. Enhanced augmented reality applications in vehicle-to-edge networks. In *ICIN*. IEEE, IEEE, 167–174.
[296] S. Zollmann, R. Grasset, T. Langlotz, W. Lo, et al. 2020. Visualization Techniques in Augmented Reality: A Taxonomy, Methods and Patterns. *IEEE Transactions on Visualization and Computer Graphics* (2020).
[297] B. Zoph and Q. Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).

Just Accepted