Received March 8, 2017, accepted April 6, 2017, date of publication April 26, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2698164

Mobile Augmented Reality Survey: From Where We Are to Where We Go

DIMITRIS CHATZOPOULOS, CARLOS BERMEJO, ZHANPENG HUANG, AND PAN HUI
Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong

Corresponding authors: Dimitris Chatzopoulos (dcab@cse.ust.hk) and Carlos Bermejo (cbf@cse.ust.hk)

This work was supported in part by the General Research Fund from the Research Grants Council of Hong Kong under Grant 26211515 and in part by the Innovation and Technology Fund from the Hong Kong Innovation and Technology Commission under Grant ITS/369/14FP.

ABSTRACT The boom in the capabilities and features of mobile devices, like smartphones, tablets, and wearables, combined with the ubiquitous and affordable Internet access and the advances in the areas of cooperative networking, computer vision, and mobile cloud computing transformed mobile augmented reality (MAR) from science fiction to a reality. Although mobile devices are more constrained computationalwise from traditional computers, they have a multitude of sensors that can be used to the development of more sophisticated MAR applications and can be assisted from remote servers for the execution of their intensive parts. In this paper, after introducing the reader to the basics of MAR, we present a categorization of the application fields together with some representative examples. Next, we introduce the reader to the user interface and experience in MAR applications and continue with the core system components of the MAR systems. After that, we discuss advances in tracking and registration, since their functionality is crucial to any MAR application and the network connectivity of the devices that run MAR applications together with its importance to the performance of the application. We continue with the importance of data management in MAR systems and the systems performance and sustainability, and before we conclude this survey, we present existing challenging problems.

INDEX TERMS Mobile augmented reality, mobile computing, human computer interaction.

I. INTRODUCTION

During the last decade, Mobile Augmented Reality (MAR) attracted interest from both industry and academia. MAR supplements the real world of a mobile user with computer-generated virtual contents. The intensity of the virtual contents and their affect on the view of the mobile user determine the reality or the virtuality, in the case of intense graphics that change the original view, of the mobile user. Figure 1 depicts the categorisation between the different versions of reality and virtuality. Real Reality is the environment of the user without the use of any device while Virtual Reality is the reality that users experience, which is unrelated with their environment and is completely generated by a computer. Mobile technology improvements in built-in cameras, sensors, computational resources and mobile cloud computing have made AR possible on mobile devices.

The advances on human computer interaction interfaces, mobile computing, mobile cloud computing, scenery understanding, computer vision, network caching and device to device communications have enabled new user experiences that enhance the way we acquire, interact and display information within the world that surrounds us. We are now able to blend information from our senses and mobile devices in myriad ways that were not possible before.

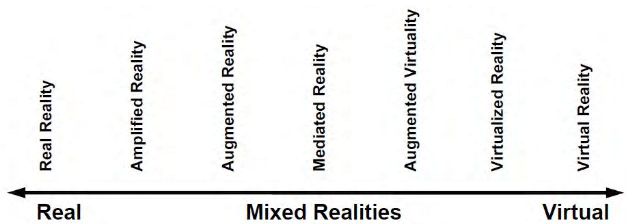


FIGURE 1. Order of reality concepts ranging from Reality (left) to Virtuality (right) as presented in [7].

Cloud infrastructure and service providers continue to deploy innovative services to breed new MAR applications. The MAR-based mobile apps and mobile advertising market is more than $732 million [1]. By considering all the definitions from various researchers in the past [2]–[6], we can conclude that MAR:

1) combines real and virtual objects in a real environment,
2) is interactive in real time,
3) registers and aligns real and virtual objects with each other, and
4) runs and/or displays the augmented view on a mobile device.

Any system with all above characteristics can be regarded as a MAR system. A successful MAR system should enable users to focus on application rather than its implementation [8]. During the last years, many case specific MAR applications have been developed with the most of them in the areas of tourism and culture and education while there is currently huge interest in MAR games. Pokemon GO,[1] for example, is a well-known MAR application that offers location-based AR mobile game experience. Pokemon GO shares many features with a previous similar MAR application, named Ingress[2] and although it gain huge popularity the first days after its release, by generating almost 2 million US dollars revenue per day, it is now loosing its popularity.[3] Some more examples are the work of Billinghurst *et al.* [9], who examined the impact of an augmented reality system on students' motivation for a visual art course and the work of Geiger *et al.* [10] who discussed location-based MAR applications.

Since the applicability of MAR is very broad, we dedicate a big part of this survey on the presentation and discussion of these cases. Due to their mobile nature, most MAR applications tend to run on mobile/wearable devices, such as smart glasses, smartphones, tablets, or even in some cases laptops. A mobile application can be categorised as a MAR application if it has the following characteristics:

Input: It considers the various sensors of the device (camera, gyroscope, microphone, GPS), as well as any companion device [11].

Processing: It determines the type of information that is going to render in the screen of the mobile device. In order to do that it may require access stored locally in the device or in a remote database.

Output: It projects its output to the screen of the mobile device together with the current view of the user (i.e. It augments the reality of the user).

AR glasses are the best option for ubiquitous mobile AR as the projected information is directly superimposed to the physical world, although their computing power is limited and, due to that, most applications remain quite basic. Smartphones are also a good option, due to their higher computing power and portability, but require the user to ''point and hold'' for being able to benefit from AR applications. Tablets, PC and laptops start to get cumbersome and limit their use to specific operations.

Due to specific mobile platform requirements, MAR suffers from additional problems such as computational power and energy limitations. It is usually required to be self-contained so as to work in unknown environments. A typical MAR system comprises mobile computing platforms, software frameworks, detection and tracking support, display, wireless communication, and data management. There are also cases where the mobile users are interacting with the application using external controllers [12]. Jain *et al.* [13] in their work in the OverLay project discuss the requirements of an MAR system to be functional.

The most widely adapted devices for AR are also the least powerful due to their high portability. Depending on the generality of an application, **(1)** the storage and **(2)** rendering capabilities of the device as well as **(3)** its connectivity to the Internet, parts of it may be executed in a cloud surrogate [14]. Vision-based applications are almost impossible to run on wearables, and very difficult on smartphones since they require capable GPUs [15], [16]. Some operations only run flawlessly on a desktop computer, or even on some dedicated server. Computation offloading solutions can therefore be used for the execution of heavy computations to a distant but more powerful device. The capabilities of the surrogate devices and accessibility vary on their type and the considered network architecture. The one extreme it can be a smartphone that works as a companion device to a smart glass device, while the other extreme it can be a virtual machine with almost infinite processing, memory and storage resources. In between there are FoG and D2D solutions.

Another important parameter is the memory and the storage requirements of the mobile applications. MAR browsers, for example, are projecting virtual objects in the view of the mobile users [17]. A virtual object can be a simple text, a two dimensional shape, a three dimensional shape or even a video. Each object can be associated with a location as well as with other objects and mobile users. The amount of the potential virtual objects is huge and this makes impossible as well as wasteful for a mobile device to store all of them locally. In such cases the MAR application is assisted by Database-as-a-Service (DBaaS) solution [18] and implements caching and pre-fetching algorithms in order to make sure that the needed virtual objects are stored locally. Figure 2 shows the basic components of a MAR system.

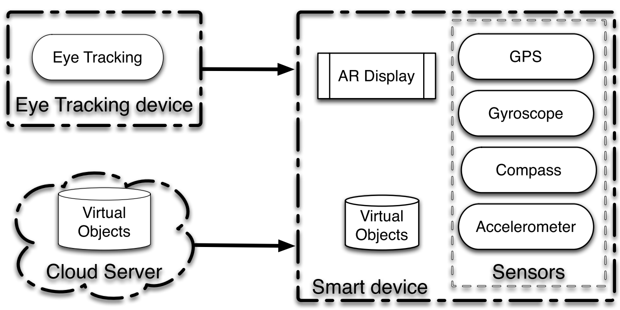


FIGURE 2. An abstract representation of the components of a MAR system. Most of the components are located in the smart device, that is responsible for the execution of the MAR application. An eye tracking device can offer better quality of experience to the mobile users since it allows them to not use their hands on the application execution. A cloud server is required for storing the virtual objects.

Furthermore, deep learning solutions have been currently utilised by MAR application developers for scenery and object recognition since it is one of the core parts of the MAR applications that needed input from the ambient environment of the mobile user. Pre-trained models are usually deployed

[1] http://www.pokemongo.com
[2] https://www.ingress.com
[3] http://expandedramblings.com/index.php/pokemon-go-statistics/

TABLE 1. Several representative MAR applications.

System	Hardware platform	Software platform	Display	Tracking	Network	Collaborative	Indoor/ outdoor	Application field
MARS	notebook computer	JABAR/ Ruby	optical see-through HMD	RTK GPS, orientation tracker	campus WLAN	multiusers	indoor, outdoor	Tourism, Navigation
ARQuake	notebook computer	Tinmith-evo5	optical see-through HMD	GPS, digital compass	cable network	single	indoor, outdoor	Entertainment
BARS	notebook computer	unknown	optical see-through HMD	inertial sensors, GPS	WWAN	multiusers	outdoor	Training
Medien. welten	PDA	Studiers-tube ES	video see-through screen	visual marker tracking	WLAN	multiusers	indoor	Education, Entertainment
MapLens	mobile phone	unknown	video see-through phone screen	nature feature tracking	WWAN	multiusers	indoor, outdoor	Tourism, Entertainment
Virtual LEGO	mobile phone	UMAR	video see-through phone screen	visual marker tracking	unknown	single	indoor	Authoring, Assembly
InfoSPOT	Apple iPad	KHARMA	video see-through pad screen	sensors, geo-reference visual markers	WWAN	single	indoor	Information Management
Pokemon Go	mobile phone	Android & iOS	video see-through screen	GPS, digital compass	Internet Access	multiusers	indoor, outdoor	Entertainment
Ingress	mobile phone	Android & iOS	video see-through screen	GPS, digital compass	Internet Access	multiusers	indoor, outdoor	Entertainment

in cloud servers and the mobile applications are imposing ad hoc queries [19]–[22].

It is easy to see that the capabilities of any device are only limited by the network access due to its core importance in both computation offloading as well as in content retrieval. Due to the potentially large amount of physical and virtual objects to process, offloaded MAR applications may require large amounts of bandwidth. Similarly, due to its real time and interactive properties, MAR is also drastically latency constrained.

In this survey, we first present the application fields of the MAR applications and some representative applications (Section II), next we discuss the advances in user interfaces and user experience evaluation (Section III). After that, we provide details about the available system components in the MAR ecosystems (Section IV). After these we discuss in detail the existing solutions regarding tracking and registration (Section V) as well as the effect of the underlying network connectivity (Section VI) and the data management (Section VII). Finally, we present the factors of the system performance sustainability (Section VIII) and the challenges (Section IX) before we conclude the survey (Section X).

II. APPLICATION FIELDS

The huge spread of mobile devices and the appearance of wearable devices and smart glasses together with advances on computer vision have given wide applicability to MAR applications. In this section we present some representative applications on the following general categories: Tourism and Navigation (Section II-A), Entertainment and Advertisement (Section II-B), Training and Education (Section II-C), Geometry Modeling and Scene Construction (Section II-D), Assembly and Maintenance (Section II-E) and Information Assistant Management (Section II-F). Last, in Section II-H we present some representative MAR applications that are listed in Table 1.

A. TOURISM AND NAVIGATION

Researchers at Columbia University built a MAR prototype for campus exploration [23], [24]. It overlaid virtual information on items in visitor's vicinity when they walked around campus. Dahne and Karigiannis [25] developed Archeoguide to provide tourists w ith interactive personalized information about historical sites of Olympia. Tourists could view computer-generated ancient structure at its original site. Fockler *et al.* [26] presented an enhanced museum guidance system PhoneGuide to introduce exhibitions. The system displayed additional information on mobile phones when visitors targeted their mobile phones at exhibits. They implemented a perception neuronal network algorithm to recognize exhibits on mobile phone.

Elmqvist *et al.* [27] proposed a three-dimensional virtual navigation application to support both path finding and highlighting of local features. A hand gesture interface was developed to facilitate interaction with virtual world. Schmalstieg *et al.* [28] constructed a MAR-based city navigation system. The system supported path finding and real object selection. When users selected a item in real world, virtual information was overlaid on view of real items.

Tokusho and Feiner [29] developed a "AR street view" system which provided an intuitive way to obtain surrounding geographic information for navigation. When users walked on a street, street name, virtual paths and current location were overlaid on real world to give users a quick overview of environment around them. Radha and Dasgupta [30] proposed to integrate mobile travel planner with social networks using MAR technologies to see feedback and share experiences. Vert and Vasiu [31] review the existing projects on

Linked Data in MAR applications for tourism. They address issues with data quality and trust, as in tourist applications the visitors rely on to visit the surroundings environment.

Furthermore, large scale integration for spatial data needs to be taken in future work. *i-Street* is an Android MAR application to read street plates from a video flow. The application will overlay information such as walking distances from POIs and targets tourist visiting a new city. Kasapakis *et al.* [32] present a MAR building recognition application, *KnossosAR*. The system provides object occlusion, and raycasting techniques to offer a better understanding for users about their surroundings. The application uses hidden markers in POIs to overlay the content, and audio as an alternative feedback for the AR experience. The goal of this dissertation [33] is to explore new ways to interact with cultural and historical sites through MAR applications. They develop an application that relies in a physical map to overlay AR content. The authors address possibly issues in case of applications that use network connectivity to provide content.

Geiger *et al.* [10] focuses on the implementation of a efficient MAR application. AR applications usually need based-location systems, inertia sensors and require heavy computational tasks in order to render the AR content in real-time. The authors provides also some insights in order to design and implement the core framework of an MAR application (Augmented Reality Engine Application, AREA). Mobile device resources and energy consumption are taken into account in AREA to design the application, and as part of their future work they want to address indoor localization to provide location-based services for MAR applications.

Jain *et al.* [13] develop a MAR application (*Overlay*) to serve as historical tour guide. They use multiple object detection and inner motion sensor to provide better indoor location accuracy and therefore, a better AR experience. They have an extensive analysis about latency and how sensor optimizations (time, full, rotation) affects the accuracy. Nartz *et al.* [34] propose the car as the device to show AR content and offer a novel alternative to navigation systems. The framework displays AR content such as path, gas station information on the user's mobile device (i.e., smartphone, PDA, tablet). The authors present a prototype of their system implemented on the car's windscreen.

They also point out some of the issues of wearable AR devices due to their size, and the need of MAR devices that merge seamlessly with the user's environment. *JoGuide* [35] is a MAR application to help tourist to recognize areas, locate POIs, and overlay the corresponding information using an Android smartphone. The authors use gelocation-based approach gathering information from users' GPS and web services such as *foursquare*.[4] Shi *et al.* [36] propose a MAR application for individual location recommendation. They use object detection and GPS data to identify the image and locate, then the application renders the corresponding information on the user's display. The experiment results show an improvement on accuracy when the application uses both location and image recognition instead of only GPS location, 60% versus 40%. Vert and Vasiu [37] present a model and an implemented prototype that integrates open and government data in a MAR tourism application. Tourist are willing to be provided with context-sensitive and dynamic information. One suitable solution is integrating linked open sources to overcome the current static-content MAR applications (i.e., applications that overlays *Wikipedia* information). This paper proposes the guidelines to implement such linked data MAR system. The authors highlight some issues during the prototype implementation regarding POIs names and position vary between sources, and the significant amount of manual effort to translate the data. This paper [38] presents a reusable MAR framework to address the repetitive and crucial task of outdoor visualization. The authors claim that the existing software does not provide a fast development environment for testing and research. Therefore, they propose a framework that will be pluggable using Object Oriented Design (OOD) techniques and modular design.

B. ENTERTAINMENT AND ADVERTISEMENT

Renevier and Nigay [2] developed a collaborative game based on Studierstube. The game supports two users to play virtual 3D chess on a real table. Piekarski and Thomas [39] proposed an outdoor single-player MAR games ARQuake. It enabled players to kill virtual 3D monsters with physical props. Cheok *et al.* [40] developed an MAR game for multiplayers in outdoor environment. They embed Bluetooth into objects to associate them with virtual objects. System could recognize and interact with virtual counterparts when players interacted with real objects. Players kill virtual enemies by physical touch other players in real world. Henrysson *et al.* [41] built a face-to-face collaborative 3D tennis game on Symbian mobile phones. Mobile phone was used as tangible racket to hit virtual ball. Hakkarainen and Woodward [42] created a similar table tennis MAR game. They proposed a color-based feature tracking approach to improve runtime performance. The game was feasible and scalable in various conditions as it required no pre-defined markers. Morrison *et al.* [43] developed a MAR map MapLens to superimpose virtual content on paper map when users targeted mobile phones at paper map. The system was used to help players complete their game. There are also many MAR applications exploited to augment advertisements.

The website [44] list several MAR-based apps for advertisement. Absolut displayed augmented animation to show consumers how vodka was made as users point their mobile phones at a bottle of vodka. Starbucks enable customers to augment their coffee cups with virtual animation scenes. Juniper research [1] estimated that MAR-based mobile apps and mobile advertising market would reach $732 million in 2014. Panayiotou [45] develops a game inspired by the platform puzzle game Lemmings for MAR systems. They use the *Vuforia* SDK to develop the AR game

[4] https://foursquare.com/

(virtual object, tracking), and Microsoft Kinect[5] to scan the 3D real environment. The participants' results in the authors' designed experiment show immersion game features. Chalvatzaras *et al.* [46] develop a MAR application for visitors to experience the historical center of a Greek city. The evaluation of their application show that MAR applications create an immersive experience, also the game interaction should be simple enough not to affect the performance and engagement of users. Kim and Kim [47] design a markerless MAR application for advertisement. The application uses *Vuforia* SDK to scan the real object and overlay to effectively convey the information of advertisement.

Olsson *et al.* [48] realize a user case study of MAR services in shopping centers. The authors conduct 16 semi-structured interview with 28 participants in shopping centers. From the experiment results, authors claim that MAR applications have great potential to provide context-sensitivity, emotions, engagement. Furthermore, the results show also some guidelines for building the next UX for MAR applications such as audio-visual feedback, readability, and accuracy of AR content. Dacko [49] discusses how and why the use of MAR applications can impact the retail market. They conduct a large-scale survey across United States smartphone users. The experiment findings show that MAR applications can add more value to the shopping experience, such as more detailed information, users' product ratings. However, designers need to be careful with the privacy settings as it is one of the most mentioned concern of the participants. For example, giving too much personal information to an application.

C. TRAINING AND EDUCATION

The Naval Research Lab (NRL) developed a MAR-based military training system [50] to train soldiers for military operations in different environments. The battlefield was augmented with virtual 3D goals and hazards which could be deployed beforehand or dynastically at run time. Traskback and Haller [51] used MAR technology to augment oil refinery training. Traditional training was conducted in classrooms or on-site when the plant was shut down for safety consideration. The system enabled on-site device-running training so that trainees could look into run-time workflow. Klopfer *et al.* [52] proposed a collaborative MAR education system for museum. Players use Pocket PCs and walkie-talkies to interview virtual characters and operate virtual instruments. To finish the task, children were encouraged to engage in exhibits more deeply and broadly. Schmalstieg and Wagner [53] built a MAR game engine based on Studierstube ES. In order to finish the task, players had to search around with cues displayed on handheld device when players pointed their handheld device at exhibits. Freitas and Campos [54] developed an education system SMART for low-grade students. Virtual 3D models such as cars and airplanes were overlaid on real time video to demonstrate concepts of transportation and animations.

[5]https://developer.microsoft.com/en-us/windows/kinect

In [55] authors study users' engagement during a collaborative AR game. Bower *et al.* [56] discuss the pedagogical potential that AR can bring to mainstream society and educational system. The paper presents a *learn by design* case studied in School of Arts, which resulted in higher creativity and critical analysis. The authors suggest that it is important for educators to keep updating their knowledge about AR technologies in order to prepare themselves and their classes for future developments. They also mentioned the use of smartphones and tablets to provide the AR scenarios in the learning process. Koutromanos *et al.* [57] realized an extensive relative literature about AR and informal and informal education environments. The paper also provides several study insights whose prove the evidence of positive outcome regarding student learning. Reference [58] is an up-to-date survey on AR in education, which includes factors such as uses, advantages, features, and effectiveness of the AR approach.

The number of AR studies has incremented significantly since 2013, the majority of these studies report that the inclusion of AR in education environments can lead to better learning performance and student engagement. Furthermore, the authors suggest as future work, additional interactive strategies to enhanced first-hand experiences, and lengthening the time span of the research AR studies. Pence [59] explains the advantages and disadvantages of the different approaches to mark real objects for AR environments, and the forthcoming opportunities for museums and libraries. Lan *et al.* [60] propose a mobile peer-assesment AR framework, in which the students can present their own work and evaluate others. Furthermore, the system also provide location-based adaptive content and how their work can be applied in real scenarios. As previous papers, the framework leans on better understanding, and critical thinking skills.

Prieto *et al.* [61] review several studies on augmented paper (i.e., documents, notebooks with fiducial markers, user input digitalization) in education scenarios. Using the notion of *class orchestration* as a conceptual tool to describe the potential integration of augmented paper in class environments. Furthermore, the analysis point out design difficulties for augmented paper and the advantages of MAR. Shuo *et al.* [62] prove real-world information based in AR can be used for education. However, the old AR system have several weakness regarding the projection such as luminance and contrast changes in real world. The authors address these issues and propose a system that can improve immersion and interaction user-AR objects. FitzGerald *et al.* [63] study AR is embedded in outdoor settings for learning processes, they also attempt to classify the key aspects of this interaction. As the authors mentioned, the AR for mobile experiences is still in its infancy. They also have another opinion against previous mentioned studies here, that it is still not straightforward to see how useful is for creating learning experiences. However, they agree that MAR can provide a better immersive experience and engagement.

Chiang *et al.* [64] propose a AR mobile learning system based on GPS location to position virtual objects in the real world in Natural Science projects. The experimental results found that the AR mobile system improved students' learning capabilities. Although, there are some constraints in the proposed system as the GPS accuracy of object positioning. Dunleavy and Dede [65] summarize existing related work about MAR in educational environments. The authors address the need to explore how AR can improve not only the learning process but other educational and pedagogic issues in the classroom.

Chang *et al.* [66] found evidence that MAR influence students to consider affective, and ethical aspects for problem solving and decision making. Bacca *et al.* [67] analyze published study findings and trends from 2003 to 2013 about MAR in educational settings. Besides the already mentioned advantages, the authors enumerate some issues with the MAR systems such as very few systems that take into account accessibility, difficulties to maintain the superposed information, and that most of the studies have used mixed evaluation methods and relative medium samples sizes (30 to 200 participants). Wu *et al.* [68] discuss how different categories of AR approaches can affect the learning process. For example, in particular AR environments the users may be overloaded by the amount of provided information and different devices to use. They also support the idea of greater samples for experiment studies and the benefits of AR in the classroom. They propose a categorization of AR approaches from instructional approaches (i.e., roles, locations, tasks). Wide the subjects with MAR techniques has to be addressed in future works, how to integrate theses new techniques into the regular school curricula. Kamphuis *et al.* [69] describe the potential of MAR for training psycho-motor skills and visualize concepts to improve the understanding of complex causality. MAR systems can improve the students enhancement inside and outside the classroom [70]. The use of books and notebooks with MAR systems can lead to a better learning process, although the lack of content creation tools is one of current problems for educators to implement MAR in educational environments. Construct3D [71] is a three dimensional geometric tool based on Studierstube [72]. They use a stereoscopic head-mounted device to provide a 3D augmented reality to interact with 3D geometric constructions in space. Chang *et al.* [73] develop a MAR system to improve art museum visitors learning effectiveness during their visit. The system locates and overlays information over current pictures, providing functions such as zooming in the virtual picture. The authors want to integrate art appreciation with MAR techniques.

However, the developed system can be used in other kind of exhibitions such as theme parks, other museums or education centers to improve visitors engagement and interaction. Radu [74] compare student learning performance in AR versus non-AR applications, they also list the advantages and disadvantages of AR in the learning process. MAR applications improve learners performance due to information can be spatio- and temporally aligned with physical objects, enhanced memory (due to immersion of AR experiences), attention is directed to relevant content, interaction and collaboration, and students motivation. However, AR applications can also have negative effects such as attention decrement to errors (i.e., 3D building in AR environment against paper-based), usability difficulties in some AR examples and ineffective classroom integration between students and students-teachers.

D. GEOMETRY MODELING AND SCENE CONSTRUCTION

Baillot *et al.* [75] developed a MAR-based authoring tools to create geometry models. Modelers extracted key points from real objects and then constructed geometric primitives from points to create 3D models. Created models were registered and aligned with real objects for checking and verification. Piekarski and Thomas [76] built a similar system for outdoor object creation. It used pinch gloves and hand tracking technologies to manipulate models. The system was specially suitable for geometrical model creation of giant objects (e.g. building) as users could stand a distance away. Ledermann and Schmalstieg [77] developed a high-level authoring tool APRIL to design MAR presentation. They integrated it into a furniture design application. Users could design and construct virtual model with real furniture as reference in the same view. Henrysson *et al.* [78] employed MAR to construct 3D scene on mobile phone in a novel way. Motions of mobile phone were tracked and interpreted to translation and rotation manipulation of virtual objects. Bergig *et al.* [79] developed a 3D sketching system to create virtual scenes for augmented mechanical experiments. Users use their hands to design experiment scene superimposed on a real drawing. Hagbi *et al.* [80] extended it to support virtual scene construction for augmented games.

E. ASSEMBLY AND MAINTENANCE

Klinker *et al.* [81] developed a MAR application for nuclear plant maintenance. The system created an information model based on legacy paper documents so that users could easily obtain related information that was overlaid on real devices. The system was able to highlight fault devices and supplied instructions to repair them. Billinghurst *et al.* [82] created a mobile phone system to offer users step-by-step guidance for assembly. A virtual view of next step was overlaid on current view to help users decided which component to add and where to place it in the next step. Henderson and Feiner [83] developed a MAR-based assembly system. Auxiliary information such as virtual arrows, labels and aligning dash lines were overlaid on current view to facilitate maintenance. A study case showed that users completed task significant faster and more accurate than looking up guidebooks. An empirical study [84] showed that MAR helped to reduced assembly error by 82%. In addition, it decreased mental effort for users. However, how to balance user attention between real world and virtual content to avoid distraction due to over-reliance is still an open problem. Webel *et al.* [85] present the need of efficient training systems for maintenance and assembly and how MAR can improve the trainer skills with appropriate methods. The authors develop and evaluate their training platform, where they highlight the need of vibrotactile feedback in this particular scenario in order to improve user interaction with the AR content. The experiment results suggest the improvement in participants skills. However, the maintenance and assembly training need to give special attention of capturing and interpretation of underlying skills. Moreover, the data captured from the system can be used to show how experts would resolve a particular training case. Furthermore, MAR opens the possibility of remote interaction expert-trainee in the field.

F. INFORMATION ASSISTANT MANAGEMENT

Goose *et al.* [86] developed a MAR-based industrial service system to check equipment status. The system used tagged visual markers to obtain identification information, which was sent to management software for equipment state information. Data such as pressure and temperature were sent back and overlaid for display on the PDA. White *et al.* [87] developed a head-worn based MAR system to facilitate management of specimens for botanists in the field. The system searched a species database and listed related species samples side-by-side with physical specimens for comparison and identification. Users slid virtual voucher list with head horizontal rotation and zoom the virtual voucher by head nodding movements. Deffeyes [88] implemented an Asset Assistant Augmented Reality system to help data center administrators find and manage assets. QR code was used to recognize asset. Asset information was retrieved from a MAMEO server and overlaid on current view of asset.

MAR also finds its markets in other fields. MAR has been used to enhance visualization and plan operations by placing augmented graphic scans over surgeons' vision field [89]. Rosenthal *et al.* [90] used the "x-ray vision" feature to look through the body and made sure the needle was inserted at the right place. MAR was also used to manage personal information [91]. Another large part of applications is AR browsers on mobile phones [92], [93]. AR browser is similar to MAR navigation application, but more emphasizes on location-based service (LBS). Grubert *et al.* [94] conducted a detailed survey about AR browsers on mobile phones.

G. BIG DATA DRIVEN MAR

Huang *et al.* [101] list some data driven examples for real application scenarios and provide a categorisation. The four proposed categories are: *(1)* Retail, *(2)* Tourism, *(3)* Health Care and *(4)* Public Services. MAR applications in Retail boost the shopping experience with more product information, the ability to change based on users emotions, and personalized content. For the case of Tourism, MAR applications use geo-spatial data to assists users on their browsing in unfamiliar environments while Health case, MAR applications aid doctors in operations and nurses in care-giving. Last, MAR applications in Public Services help citizens in their everyday routine. In all these for cases, these applications are functional only because they have access to big data sources. In Figure 3 we present an example were big data is used in the visualisation of the wind movement. It is easy to see that this detailed representation requires huge amount of data since it can not be described through modeling.

FIGURE 3. Visualization of a numerical flow field with real buildings makes the influence of the building on wind movement easily understood. (source: http://emcl.iwr.uni-heidelberg.de/researchsv.html)

H. REPRESENTATIVE MAR SYSTEMS

Although there are many MAR systems, as discuss earlier in this section, we use this subsection to discuss a representative sample of them in more detail. Table 1 contains their characteristics in a collective way.

1) MARS [23], [24]

is both indoor and outdoor MAR system developed by a team at Columbia University. They have built several iterations of the prototype and developed a series of hardware and software infrastructures. The system comprises a backpack laptop, a see-through head-worn display and several input devices including stylus and trackpad. An orientation tracker and RTK GPS are used to obtain pose information. A campus guide system has been developed based on MARS. Visitors are able to obtain detailed information overlaid on items in their current view field. They can also watch demolished virtual buildings on their original sites. It supports virtual menus overlaid on users view field to conduct different tasks. The system can also be used for other applications such as tourism, journalism, military training and wayfinding.

2) ARQuake [39]

is a single-player outdoor MAR games based on popular desktop game Quake. Virtual monsters are overlaid on current view of real world. Player move around real world and use real props and metaphors to kill virtual monsters. Real buildings are modeled but not rendered for view occlusion only. The game adopts GPS and digital compass to track player's position and orientation. A vision-based tracking method is used for indoor environments. As virtual objects may be difficult to recognize from natural environments at outdoor environments, system have to run several times to set a distinguishable color configuration for later use.

3) BARS [50]

is a battlefield augmented reality system for soldiers training in urban environments. Soldiers' perceptions of battlefield environment are augmented by overlaying building, enemies and companies locations on current field of view. Wireframe plan is superimposed on real building to show its interior structures. An icon is used to report location of sniper for threat warning or collaborative attacking. A connection and database manager is employed for data distribution in an efficient way. Each object is created at remote servers but only simplified ghost copy is used on clients to reduce bandwidth traffic. The system requires a two-step calibration. The first is to calculate mapping of result from a sensing device to real position and orientation of sensors; the second is to map sensing unit referential to viewpoint referential of user's observation.

4) Medien.welten [53]

is a MAR system that has been deployed at Technisches Museum Wien in Vienna. The system is developed based on Studierstube ES. A scene graph is designed to facilitate construction and organization of 3D scenes. Total memory footprint is limited to 500k to meet severe hardware limitation on handheld devices. Game logic and state are stored in a XML database in case of client failure due to wireless single shielding. Medien.welten enables players to use augmented virtual interface on handheld devices to manipulate and communicate with real exhibits. With interactive manipulation, players gain both fun experience and knowledge of exhibits.

5) MapLens [43], [95]

is an augmented paper map. The system employed mobile phones' viewfinder, or "magic lens", to augment paper map with geographical information. When users view a paper map through embedded camera, feature points on paper map are tracked and matched against feature points tagged with geographical information to obtain GPS coordinates and pose information. GPS coordinates are used to search an online HyperMedia database (HMDB) to retrieve location-based media such as photos and other metadata, which are overlaid on paper map from current pose. Augmented maps can be used in collaborative systems. Users can share GPS-tagged photos with others by uploading images to HMDB so that others can view new information. It establishes a common ground for multiple users to negotiate and discuss to solve the task in a collaborative way. Results show that it is more efficient than digital maps.

6) VIRTUAL LEGO [78]

uses mobile phone to manipulate virtual graphics objects for 3D scene creation. The motion of mobile phone is employed to interact with virtual objects. Virtual objects are fixed relative to mobile phone. When users move their mobile phone, objects are also moved according to relative movement of mobile phones to the real world. In translation mode, the selected object is translated by the same distance as mobile phone. Translation of mobile phone is projected onto a virtual Arcball and converted as rotation direction and angle to rotate virtual object. The objects are organized in a hierarchical structure so that transformation of a parent object can be propagated to its sub-objects. A multiple visual markers tracking method is employed to guarantee accuracy and robustness of mobile phone tracking. Results show that the manipulation is more efficient than button interface such as keypad and joypad, albeit with relative low accuracy.

7) InfoSPOT [96]

is a MAR system to help facility managers (FMs) access building information. It augments FMs' situation awareness by overlaying device information on view of real environment. It enables FMs to fast solve problems and make critical decisions in their inspection activities. The Building Information Modeling (BIM) model is parsed into geometry and data parts, which are linked with unique identifiers. Geo-reference points are surveyed beforehand to obtain accurate initial registration and indoor localization. The geometry part is used to render panoramas of specific locales to reduce sensor drift and latency. When FMs click virtual icon of physical object on IPad screen, its identifier is extracted to search data model to fetch information such as product manufacture, installation date and life of product.

8) POKEMON GO & INGRESS

are two MAR games both developed by Niantic, Inc. In both of them the mobile users augment their view though the screen of the mobile device, are multiuser games, require Internet access and use GPS for positioning. In both games, a cloud server is used to locate virtual objects in the form of Pokemons or events and the users compete with each other to arrive first in these locations. Also, the users are able to set up battles with each other and earn others' winnings.

Table 1 lists system components and enabling technologies of aforementioned MAR applications. Early applications employed backpack notebook computer for computing tasks. External HMDs were required to provide optical see-through display. As mobile devices become powerful and popular, numerous applications use mobile devices as computing platforms. Embedded camera and self-contained screen are used for video see-through display. Single tracking methods have also been replaced with hybrid methods to obtain high accurate results in both indoor and outdoor environments. Recently applications outsource computations on server and cloud to gain acceleration and reduce client mobile device requirements. With rapid advances from all aspects, MAR will be widely used in more application fields.

III. UI/UX

User Interface and experience is a key factor in users engagement for MAR. The difference between good and bad implementation can make great changes in users' opinion about the AR content and interactivity. Although users center design still holds the core in designing User Interfaces (UI), it is not enough for applications to provide usability. User eXperience (UX) involves users' behaviour, emotions towards a specific artifac, and needs to be consider in any current application design for MAR.

Wang *et al.* [97] propose a framework to provide an experiment sandbox where different stakeholder can discuss the design of future AR applications. The paper [98] contains design approaces for advertisement industry for improving UX. The results from an experimental study will provide future guidelines to follow in following UI designs: the interface should be fun and usable, the UI should be quick and responsive, design to enhanced utility, the 3D design of AR content plays an important role on users perception. Irshad and Awang Rambli [99] propose a multi-layered conceptual UX framework for MAR applications. They present the important products aspects that need to be addressed while designing for MAR.

The goal of this framework is to be the guideline for design and evaluation of future MAR applications such as presentation, content, functionality, users time experience during actions, specific context (i.e., physical), and experience being invoked. Ubiquitous interface interaction (Ubii) is a novel interface system that aims to merge the physical and the virtual world with hand gesture interactions. Ubii is a innovative smart-glass system that provides free-hand user experience to interact with physical objects. For example, the user can send a document to a printer with a simple pinch hand gesture from the PC (where the document is) to the printer using smart-glasses to visualize the action. The Master Thesis [100] aims to provide a better presentation of information for indoor navigation systems using MAR technologies. The experiment result suggest that providing information on a map is not sufficient for indoor navigation, for example showing AR content in similar manner than modern GPS may improve the user experience. This paper [102] presents novel key ideas to develop a framework for context and user-based context, to improve user experience immersion. Context immersion can be define as user awareness of real context through interactions with the AR content. The paper aims to get a better understanding of UX in MAR applications, and it is constructed based in three context dimension: time and location-based tracking (i.e., GPS); object-based context immersion (i.e., object recognition); user-based context immersion (i.e., multiple users communication, interaction).

These three dimensions and the insights from this paper's empirical results need to be considered when designing MAR applications. Hürst and Van Wezel [103] present two experiments to explore new interaction metaphors for MAR applications. The experiments evaluate canonical operations such as translation, rotation, and scaling with real and virtual objects. One of the major problem we face with AR content is the lack of feedback when touching and moving the virtual objects.,to overcome this issue the framework provides a clear description and visualization of a particular interaction (i.e., overlay information, color fingertips). Other interesting outcome from the experiment is the preference for one finger gesture to rotate the virtual objects instead of using two fingers as we would do in the real world scenario. Therefore, designer need to focus not only in translate real world interactions into the virtual world, but to analyse each interaction to find the best user experience. Most of the MAR applications simulate real interactions in the augmented world without considering the manifold contextual factors of their use [104] . In this workshop the authors suggest some guidelines to follow such as *designing with reality and beyond* to provide usable and interactive experiences with the AR content, *adaptative AR* situate the content with relation to users position and change dynamically according to it, *rapid prototyping tools* to improve applications development.

MAR application design face several challenges [105] that include: Discoverability, how to find the MAR services; Interpretability, value that provides MAR services; Usability, UI/UX features and MAR application-user interaction. The authors enumerate the design approaches to achieve a good MAR experience. Developers of MAR applications need to evaluate the usability and effectiveness of their proposed application. There are extensive work on how to translate real interactions into the augmented world. However, the designer need also to consider other interactions besides the real world ones in order to provide a better UX. Furthermore, the authors develop a prototype based on the insights of other research papers (Friend Radar) and evaluate the MAR application according to concept and feature validation, usability testing and UX evaluation. To summarize, the authors try to respond to the question: how can we develop MAR applications that are usable and realistic? AR and MAR experiences can sometimes look *messy* because of the multi-layer nature (i.e., UI and information) of the applications [106]. Within this AR/MAR paradigm, designer have the chance to redefine the relationship between information and the physical world. Due to the pervasive nature of mobile devices many of the MAR applications provide a social layer content (i.e., Facebook, Twitter), so the users can interact with other users of the platform. Several studies suggest that the society is shifting the culture's aesthetics due to MAR/AR applications. We need to focus on how we can have better designs and improved information using this new paradigm, MAR/AR. The authors in this paper propose a MAR Trail Guide as an example, where the users navigate from map to displayed AR content. Ahn *et al.* [107] propose the development of AR content for MAR applications in HTML5 to provide a clean separation between content and application logic. They follow a similar approach to Wikitude ([93]), so the HTML5 application can be cross-platform compatible. However, the content structure of ARchitect (Wikitude) does not rely on HTML. The AR content is contained in HTML. For example, a physical entity is associated with an URI, which returns metadata in HTML format, image and feature extracted from the image. CSS is used in this system to

TABLE 2. MAR application design guideline.

Guideline	Description
Use the context for providing content	Geolocation, object detection to provide corresponding information
Deliver relevant-to-the-task content	Clear and consistent UI; personalized, and dynamic changing content. Besides, the addition of the social aspect (i.e., Facebook, Twitter, collaborative environments) improves the user experience and enhancement with the MAR application
Provide useful interactions with the AR content	Provide information related to the product or object that empower the user interaction and experience
Inform about content privacy	Inform the user how the information collected about the users' device will be used. Besides, provides privacy awareness for the AR content
Provide feedback about the infrastructures behaviour	Provide different configurations of an application regarding quality, resource requirements. Update the interaction according to users' interactions
Support procedural and semantic memory	Focus the UI design in usability in order to make easier for non-experienced users to interact with the AR applications. Furthermore, some of the AR content interactions do not need to be an extrapolation of real world gestures (i.e., some users prefer other approaches to interact with AR objects and content)

provide augmented HTML elements onto the physical world. To demonstrate the feasibility of author's proposal, they develop an AR Web Browser to evaluate the performance. Ventä-Olkkonen *et al.* [108] conduct a field study with 35 participants, where they test two different UI approaches for MAR.

The first UI design is a standalone MAR application where the AR content is displayed over the physical world; and the second approach uses 3D models (Virtual) to represent the real world. The combination of AR content and real physical elements is more appealing to the participants as it provides a more realistic experience. However, the 3D model has some advantages in situations where there are unimportant objects in user's view. This paper [109] analyse the findings of a user case study of last UI MAR applications. The experiment consists in a cross sectional survey to evaluate the current augmented reality market services. Most of the participants have found the experience with current MAR applications lively and playful, the UI is consistent and clear. However, the connection between the application and the product is not relevant, which means that MAR applications need to empower the users and not just user AR content as a trendy visualization tool to inform users about a product. This paper [110] present a concept framework for UX of MAR applications. This framework guide the developers to design and evaluate the MAR components to achieve a good UX. The AR content should be interactive, relevant in order to provide good information to the user. The application should give appropriate feedback to user's actions and personalization of the content. User friendly menus should also be considered by developers. Finally, intuitive, simple and interactive designable elements (UI). Besides, the MAR application should also evoke emotional (i.e., love, fear, desire), aesthetic (i.e., beauty of use) experience, and lastly experience of meaning (i.e., luxury, attachment).

Khan *et al.* [111] propose a large-area interactive mixed reality system, where users can experience the same shared events simultaneously. The system does not need to have pre-defined markers, and it provides shared AR content for multiple users providing a unique collaborative experience. The experiment results show that the mixed reality experience in a group enhanced the interactivity. Kourouthanassis *et al.* [112] present a set of interactions design principles for MAR applications. The design principles are centered in ensuring the UX in MAR ecosystems and are the following: *(1)* Use the context for providing content (i.e., location). *(2)* Deliver relevant-to-the-task content (i.e., filtering and personalization of AR content). *(3)* Inform about content privacy. *(4)* Provide feedback about the infrastructure behaviour (i.e., provide different configurations of an application regarding quality, resource requirements). *(5)* Support procedural and semantic memory (i.e., making it easy for the user to learn how to use the system). The paper [113] presents the challenges for AR wearable devices to show adequate information, feedback, and proposes a framework that adapts MAR applications to an individuals environmental and cognitive state in real time. MAR applications need to be responsive and able to provide the right feedback/interaction according to environmental conditions, and users cognitive states (which changes with sleep, nutrition, stress, and even time of day). Huang *et al.* [101]describe some future concepts for AR and MAR applications. *(1)* Data Visualization with AR and MAR we can provide a landscape of data where the users can interact with. AR ecosystem opens a new approach to understand complex data structures and better for users to manage. *(2)* User interaction; AR can be a very usable canvas to visualize big data sets as we can see from movies such *Avatar or Minority Report*. Figure 4 shows the UIs of these two movies.

IV. SYSTEM COMPONENTS

Despite the number of projects and research have been done, many of the existing AR systems rely in platforms that are heavy and not practical for the mobile environment [119]. Krevelen *et al.* [120] also mention the lack of portability in some AR systems, due to computing and energy

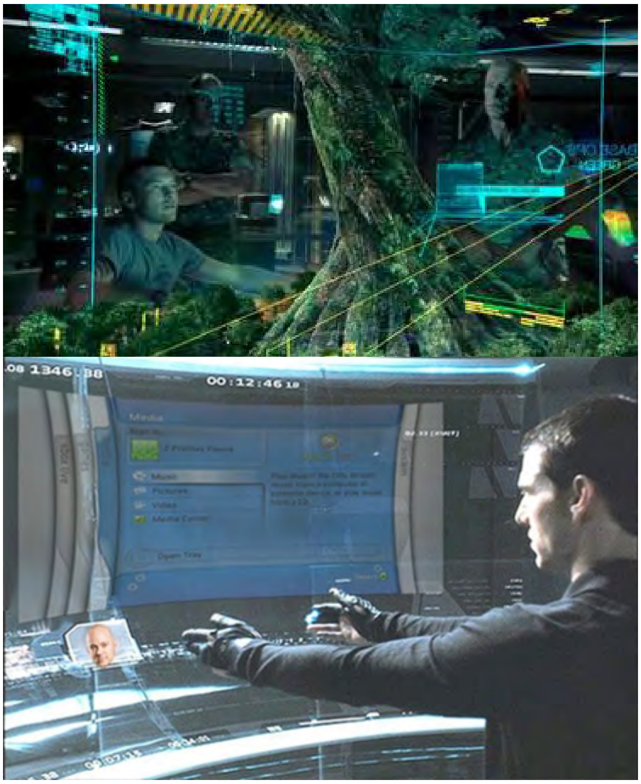


FIGURE 4. User Interfaces in Avatar (top) and Minority Report (bottom).

power constrains. Furthermore, there are technical challenges in the computer vision field such as depth perception, that has been improved in last products implementations (i.e., *Smart-glass, Microsoft Kinect*). Besides the hardware, the UI must also follows some guidelines to provide a good user experience. Last but not least, the social acceptance of this new paradigm might be more challenging than expected, we have a good example with Google-Glass. We use this Section to present existing mobile platforms in Section IV-A, the software frameworks in Section IV-B and the displays in Section IV-C.

A. MOBILE COMPUTING PLATFORMS

The high mobility of MAR requires it to provide services without constraining users' whereabouts to a limited space, which needs mobile platforms to be small and light. Recent years, we have seen significant progress in miniaturization and performance improvement of mobile computing platforms.

1) NOTEBOOK COMPUTERS

Notebook computers were usually used in early MAR prototypes [40], [121], [122] as backpack computing platforms. Comparing to consumable desktop computers, notebook computers are more flexible to take alongside. However, size and weight is still the hurdle for wide acceptance by most users. Since notebook computers are configured as backpack setup, additional display devices such as head mounted displays (HMDs) are required for display. Notebook screen is only used for profiling and debugging.

2) PERSONAL DIGITAL ASSISTANTS (PDAs)

PDAs were an alternative to notebook computers before emergence of other advanced handheld PCs. Several MAR applications [86], [123]–[128] configured PDAs as mobile computing platforms, while others [115], [129], [130] outsourced the CPU-intensive computations on remote servers to shrink PDAs as thin clients for interaction and display only. PDAs have problem of poor computational capability and absence of floating-point support. Small screen also limits the view angle and display resolution.

3) TABLET PERSONAL COMPUTERS (TABLET PCs)

Tablet PC is a personal mobile computer product running Windows operating systems. Large screen size and multi-touch technology facilitate content display and interactive operations. Many MAR systems [53], [131]–[135] were built on Tablet PCs. In addition to expensive cost, Tablet PCs are also too heavyweight for long-time single-handed hold. [124]

4) ULTRA MOBILE PCs (UMPCs)

UMPCs have been used for several MAR applications [29], [117], [136]–[139]. UMPCs are powerful enough to meet computing requirements for most MAR systems, but they are designed for commercial business market and high price impedes their widespread. Besides, They have similar problems as PDAs due to limited screen size.

5) MOBILE PHONES

Since the first MAR prototype on mobile phone [118], mobile phones have achieved great progress in all aspects from imbedded cameras, built-in sensors to powerful processors and dedicated graphics hardware. Embedded camera is suitable for video see-through MAR display. Built-in sensors also facilitate pose tracking. Many MAR applications [30], [41], [43], [78], [95], [140]–[147] were built on mobile phones. Mobile phones have become predominant platforms for MAR systems because of minimal intrusion, social acceptance and high portability. However, despite rapid advances in mobile phones as computing platforms, their performance for real-time applications is limited. The computing power is still equivalent to a typical desktop computer ten years ago [146]. Most mobile phones are equipped with relatively slow memory access and tiny caches. Build-in camera has limitations of narrow field-of-view and white noise. Accelerometer sensors are too noisy to obtain accurate position. Magnetometers are also prone to be distorted by environmental factors.

6) AR GLASSES

AR glasses leverage the latest advances in mobile computing and projection display to bring MAR to a new level. They supply a hands-free experience with lest device intrusion. AR glasses work in a way that users do not have to look down at mobile devices. However, there is controversy whether they are real MAR or not as current applications on AR glasses supply functions which are irrelative to real world content

FIGURE 5. Several mobile devices used in MAR applications. Left: Notebook computer [114]; Middle: PDA [115], Tablet PC [116], UMPC [117], and mobile phone [118]; Right: Google Glass [91].

TABLE 3. A list of different MAR computing platforms. Their characteristics and performance depends on products of different vendors.

Platforms	Computing Power	Rendering Power	Floating-point Support	User Interface	Portability	Endurance (hour)
Notebook computer	high	high	yes	keyboard/mouse	low	2∼4
PDA	low	low	no	keyboard/stylus	high	4∼6
Tablet PC	medium	medium	yes	stylus/touch	medium	5∼8
UMPC	medium	medium	yes	keyboard/dialkey/ stylus/touch	medium	4∼8
Mobile phone	low	low	no[a]	keyboard /touch	high	5∼8
AR glass	low	medium	unknown	voice/touch	high	∼6[b]

[a]Presently only a few smartphones support floating-point calculation.
[b]Google Glass.

and require no tracking and alignment. We regard it as MAR because facial recognition and path finding are suitable for AR glasses and will emerge on AR glasses in near future. In addition, from a much general point of view, AR glasses work as user interface to interact with real world analogous to most MAR systems. Currently Google Glass [91] is still the default target for MAR application research, although there are other new developments such as MAD Gaze[6] and the promising Microsoft HoloLens[7] It supports features including taking picture, searching, sending message and giving directions. It is reported to be widely delivered later this year, whereas high price and absence of significant applications may hinder its popularity to some extent.

Google Glass [91] is a wearable AR device developed by Google. It displays information on glass surface in front of users' eyes and enables users to control interface with natural language voice commands. Google Glass supports several native functions of mobile phones such as sending messages, taking pictures, recording video, information searching and navigation. Videos and images can be shared with others through Google+. Current product use smartphones as network transition for Internet access. As it only focuses on text and image based augmentation on a tangible interface, it does not require tracking and alignments of virtual and real objects.

Figure 5 shows some typical mobile devices used in MAR applications. Since most computing platforms are not designed for MAR use, specific tradeoffs between size, weight, computing capability and cost are made for different users and markets. Table 3 compares mobile platforms in terms of features that may concern.

[6]http://www.madgaze.com
[7]https://www.microsoft.com/microsoft-hololens/en-us

B. SOFTWARE FRAMEWORKS

It is complicated and time-consuming to build a MAR system from scratch. Many software frameworks have been developed to help developers focus on high-level applications other than low-level implementations. In this section we discuss a representative subset of the existing frameworks and we present them collectively in Table 4.

1) STUDIERSTUBE ES

Studierstube [72] was developed by the Institute of Computer Graphics in Vienna University of Technology. Reitmayr and Schmalstieg migrated it to mobile platforms as a sub-branch Studierstube ES [148], [149]. Studierstube ES was rewritten from scratch to leverage newly graphics APIs for better rendering capability. The system support various display devices and input interfaces. It use OpenTracker [150] to abstract tracking devices and their relations. A network module Muddleware [151] was developed to offer fast client-server communication services. A high-level description language Augmented Presentation and Interaction Language (APRL) [77] was also provided to author MAR presentation independent of specific applications and hardware platforms. Many MAR applications and

TABLE 4. Comparisons of different MAR software frameworks.

Software	Programming Language	Rendering Language	Auxiliary Tools	Tracking & Positioning	Device Support
Studierstube ES	C++	OpenGL/ES	Authoring tools (APRIL)	ARToolkitPlus [162]	Windows phone / Android
Wikitude	HTML & Javascript	Unity 3D	Cross-platform deployment		Windows phone / Android / iOS
Nexus	unknown	unknown	AR language (AWML)	external sensor system	Portable computers / handheld devices
UMAR	Web scripts	OpenGL ES[a]	no	ARToolkit [154]	Symbian mobile devices
Tinmith-evo5	C++	OpenGL	no	OpenTracker [150]	Portable computer
DWARF	C++ / Java	VRML/ OpenGL	Profiling/ debugging tools	self-contained	Portable computers / PDA
KHARMA	KML& Web scripts	OpenGL ES	Authoring tools (KML)	GeoSpots [161]	handheld devices
ALVAR	C++	third-party graphical libs	camera calibration, basic filters	ARToolkit [154]	Portable computers / handheld devices
CloudRidAR	C++	third-party graphical libs	no	OpenCV	Android
ARTiFICe	C# & C++	Unity 3D	multiuser support	Kinect	Android & iOS
AndAR	Java	OpenGL	no	ARToolkit [154]	Android
DroidAR	Java	OpenGL	no	self-contained	Android
GRATF	C♯	Direct3D	Prototyping /debugging tools	glyph recognition [163]	unknown

[a]It is uncovered in current version, but it is reported to be OpenGL ES in next version.

prototypes [53], [124], [127], [143], [147] were developed with Studierstube ES. With about two-decade persistent development and maintenance, it became one of most successful MAR frameworks. Currently Studierstube ES is only available for Windows phones and Android platforms.

2) WIKITUDE [93]

is a LBS-based AR browser to augment information on mobile phones. It is referred as "AR browser" due to its characteristic of augmentation with web-based information. Wikitude overlays text and image information on current view when users point their mobile phones to geo-located sites. Wikitude combines GPS and digital compass sensors to track pose tracking. Contents are organized in KML and ARML formats to support geographic annotation and visualization. Users can also register custom web services to get specific information.

3) NEXUS

Nexus [152] was developed by University of Stuttgart as a basis for mobile location-aware applications. It supported spatial modeling, network communication and virtual information representation. The architecture was structured in three layers [153], which were used for client devices abstraction, information uniform presentation and basic function wrapper. A hierarchical class schema was dedicated to prepresent different data objects. It supported both local and distributed data management to offer uniform access to real and virtual objects. An adaptive module was developed to ensure the scalability in different application scenarios. Nexus prevailed over other platforms in terms of stability and portability.

4) UMAR

UMAR [140] was a conceptual software framework based on client-server architecture. It was delicately designed to perform as much as possible on the client side to reduce data traffic and over-dependence on network infrastructure. UMAR imported ARToolkit [154] onto mobile phones for visual tracking. A camera calibration module was also integrated to boost accuracy. UMAR was only available to the Symbian platform. Besides, it did not support collaborative MAR applications.

5) Tinmith-evo5

Tinmith-evo5 [155], [156] was an object-oriented software framework developed by Wearable Computer Lab at the University of South Australia. Data flow was divided into serial layers with sensor data as input and display device as output. All objects in the system were allocated in an object repository to support distributed, persistent storage and runtime configuration. Render system was based on OpenGL and designed to support hardware acceleration. Several MAR applications and games [39], [76], [157] were developed with Tinmith-evo5.

6) DWARF

DWARF [158]–[160] was a reconfigurable distributed framework. A task flow engine was designed to manage a sequence of operations that cooperated to finish user tasks. The Common Object Request Broker Architecture (CORBA) was used to construct a peer-to-peer communication infrastructure and manage nodes servers. It was also employed to create wrappers for third-party components. A visual monitoring and debugging tool was developed for fast prototyping.

DWARF has been used to create several MAR applications including Pathfinder [158], FIXIT for machine maintenance and SHEEP for collaborative game [160].

7) KHARMA

KHARMA [161] was developed by GVU of Georgia Institute of Technology. It was an open architecture based on KML, a type of XML for geo-referenced multimedia description, to leverage ready-to-use protocols and content delivery pipelines for geospatial and relative referencing. The framework contained three major components: channel server to deliver multiple individual channels for virtual content, tracking server to provide location-related information and infrastructure server to deliver information about real world. Irizarry *et al.* [96] developed InfoSPOT system based on KHARMA to access building information. KHARMA supported hybrid multiple sources tracking to increase accuracy, but it was only suitable for geospatial MAR applications.

8) ALVAR

ALVAR [123] was a client-server based software platform developed by VTT Technical Research Center of Finland. Virtual contents rendering and pose calculation could be outsourced to server to leverage powerful computing and rendering capabilities. Images were then sent back to client and overlaid onto captured images on client for display. It offered high-level tools and methods for AR/MAR developments such as cameral calibration, Kalman filters and markers hiding. ALVAR supported both marker and markerless based tracking as well as multiple markers for pose detection. It was designed to be flexible and independent of any graphical and other third-part libraries except for OpenCV, so it could be easily integrated into any other applications. ALVAR was used to construct various MAR applications such as maintenance [164], plant lifetime management [165] and retail [166].

9) CloudRidAR

CloudRidAR is a cloud-based architecture for MAR [167] in order to help developers in the heavy task of designing an AR system. In cases of low requirements for rendering the AR content, *CloudRidAR* provides a local rendering engine,; for large-scale scenarios there is a cloud rendering subsystem. The user interaction is parametrized in the local device and upload to the cloud.

10) ARTiFICe

ARTiFICe [168] is a powerful software framework to develop collaborative and distributed MAR applications. The application allows collaboration between multiple users, either by focusing their mobile device to the same physical area, or showing on the device the same AR content on different physical scenarios. ARTiFICe can be implemented seamlessly in several platforms such as mobile, desktop and immersive systems, which provides 6DOF-input devices.

11) OPEN SOURCE FRAMEWORKS

Besides efforts from academic laboratories, there are several open source frameworks from developer communities. AndAR [169] is an open project to enable MAR on Android platforms. The project is still at its early age and it is only tested on very few mobile phones. DroidAR [170] is similar to AndAR but supports both location-based MAR and marker-based MAR applications. GRATF [171] is glyph recognition and tracing framework. It provides functions of localization, recognition and pose estimation of optical glyphs in static images and video files. Most open source frameworks are still under development.

Table 4 gives a comparison of several software frameworks. An ideal software framework should be highly reusable and independent of hardware components and applications. It should be used for various scenarios without reprogramming and modifications. Current frameworks are far from satisfactory for the requirements. It is difficult to abstract all hardware components with a uniform presentation, not to mention that hardware is developing.

C. DISPLAY

1) OPTICAL SEE-THROUGH DISPLAY

In optical see-through display, virtual contents are projected onto interface to optically mix with real scene. It requires the interface to be semi-transparent and semi-reflexive so that both real and virtual scenes can be seen. A head tracker is used to obtain users' positions and orientations for content alignment. Optical see-through display was used in early MAR applications [2], [172]. It enables users to watch real world with their natural sense of vision without scene distortion. The major problem is that it blocks the amount of light rays from real world and reduces light. Besides, it is difficult to distinguish virtual contents from real world when background environment is too bright.

2) VIDEO SEE-THROUGH DISPLAY

Video see-through display has two work modalities. One is to use HMD devices to replace user eyes with head-mounted video cameras to capture real world scene. Captured video is blended with computer-generated content and then sent to HMD screen for display. A head tracker is used to get users position and orientation. This mode is similar to optical see-through display and has been used in early MAR applications [40], [118], [173]. The other mode works with camera and screen in handheld devices. It uses the embedded cameras to capture live video and blend the video with virtual information before displaying it on the screen. This mode is predominant in applications with handheld devices. The former mode obtains better immersion experience at the cost of less mobility and portability. Compared to optical see-through display, mixed contents are less affected by surrounding conditions in video see-through display, but it has problems of latency and limited video resolution.

3) SURFACE PROJECTION DISPLAY

Projection display is not suitable for MAR systems due to its consumable volume and high power requirements. With recent progress of projectors in miniaturization and low power consumption, projection display finds its new way in MAR applications. Surface projection displays virtual contents on real object surface rather than display mixed contents on a specific interface. Any object surface, such as wall, paper and even human palm, can be used as interface for display. It is able to generate impressive visual results if real surface and virtual contents are delicately arranged. Pico-projectors have already been used in several MAR applications [174], [176]. Laser projector, a variation of traditional projector, has been exploited for spatial AR (SAR) applications [175]. It has many advantages including self-calibration, high brightness and infinite focal length. Since virtual information is projected to any arbitrary surface, surface projection display requires additional image distortion to match real and virtual projectors for content alignment [177].

FIGURE 6. Several display ways in MAR. Optical see-through display (Left): Sony Glasstron LDI-D100B and MicroOptical Clip-on; Video see-through display (Middle): two camera mounted HMD of UNC, mobile phone [88] and PDA [53]; Surface project display (Right): mobile camera projector [174] and laser projector [175].

Figure 6 illustrates several display devices for MAR. Jannick *et al.* [178] gave a detailed comparison of HMD-based optical and video see-through displays in terms of field of view, latency, resolution limitation and social acceptance. Optical see-through display is not often used in recent MAR applications due to sophisticated requirement of projectors and display devices, whereas Google Glass [91] proves that it is also suitable for wearable and MAR systems with micro laser projector.

V. TRACKING AND REGISTRATION

Tracking and registration is the process to evaluate current pose information so as to align virtual content with physical objects in real world. There are two types of tracking and registration: sensor-based and vision-based. Sensor-based methods employ inertial and electromagnetic fields, ultrasonic and radio wave measure and calculate pose information; vision-based methods estimate gesture information from point correspondent relationships of markers and features from captured images or videos.

A. SENSOR-BASED METHODS

According to work modalities, sensor-based methods can be divided into inertial, magnetic, electromagnetic and ultrasonic categories. For simplification, we also categorize inferred-based tracking as a type of electromagnetic method in this paper.

1) INERTIAL-BASED

Many inertial sensors output acceleration, which is integrated twice over time to obtain position and angle. Inertial-based method is able to work under most conditions without range limitation or shielding problem. Many MAR applications [23], [87], [179]–[181] use inertial sensors to get user pose information. It has problem of rapid propagation of drift due to double integration and jitters from external interference. Several methods have been proposed to improve accuracy. For example, jitter was suppressed with complementary Kalman filter [182]. In [183], drift error was minimized by taking relative measurements rather than absolute measurements.The method required a periodic re-calibration and prior knowledge of initial state to get absolute pose in a global reference frame.

2) MAGNETIC-BASED

Magnetic tracking uses earth magnetic field to get orientation. It combines with other position tracking methods to obtain six degree of freedom (6DOF). Many MAR applications [2], [184]–[186] use it to track orientation. Magnetic-based method has problem of interference by ambient electromagnetic fields. It is apt to be distorted in surroundings full of metal shields such as steel and concrete skeletons. Chung *et al.* [187] employed a server that contained magnetic fingerprint map to improve accuracy at indoor area. Sun *et al.* [188] aggregated ceiling pictures as orientation references to correct original outputs. The method achieved 3.5 times accurate improvement. As it required ceiling patterns for tracking, the method was only suitable for indoor use.

3) ELECTROMAGNETIC-BASED

Electromagnetic methods track position based on time of arrival (TOA), received signal strength indicator (RSSI) or phase difference of electromagnetic signals. There are several electromagnetic-based tracking methods in MAR field. GPS method is widely used in numerous outdoor MAR applications, but it has problem of single shielding in urban canyons and indoor environments. Besides, plain GPS is too coarse to use for accurate tracking. Some works used differential GPS [25], [173] and real-time kinematic (RTK) GPS [23] to improve accuracy but they require locations of base stations. Sen *et al.* [189] leveraged the difference of Wi-Fi signal attenuation blocked by human body to estimate position.

TABLE 5. A table compares several sensors for tracking in MAR.

Type	Sensors	Coverage	Accuracy	Indoor / outdoor
inertial	accelerometer	anywhere	0.01m	indoor / outdoor
	gyroscopes	almost anywhere[a]	0.2 deg.	indoor / outdoor
magnetic	magnetometers	almost anywhere[a]	0.5 deg.	indoor / outdoor
electro-magnetic	GPS	almost anywhere[b]	10m∼15m	outdoor
	differential GPS	almost anywhere[b]	1m∼3m	outdoor
	RTK GPS	almost anywhere[b]	∼0.01m	outdoor
	Wi-Fi	∼90m	2.5m	indoor
	RFID	20m∼100m	1m∼3m	indoor
	UWB	10m∼100m	0.15m∼0.3m	indoor
	infrared	∼6m	0.03m∼0.1m	indoor
	Bluetooth	∼10m	0.1m∼10m	indoor
ultrasonic	ultrasonic	10m	∼0.01m	indoor

[a]blind regions including environments full of strong ambient electromagnetic fields and metal shields.
[b]blind regions including urban canyons, indoor surroundings and places out of LoS.

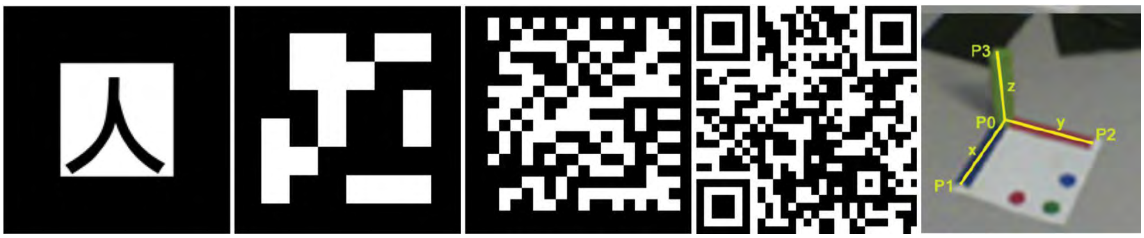


FIGURE 7. Some visual markers used for tracking. From left to right: template markers [154], BCH markers [53], DataMatrix markers [143], QR barcode markers [88], and 3D paper marker [118].

It required communication parameters and wireless maps beforehand. Wi-Fi tracking has problem of poor accuracy and tedious offline training to construct wireless map [8].Ultra Wideband (UWB) is able to obtain centimeter-level accurate results. It has been used in many systems [121], [190]–[192]. The method has drawbacks of high initial implementation cost and low performance. Radio Frequency Identification (RFID) depends on response between RFID readers and RFID tags for tracking [8], [126], [193]. As each target requires RFID tag for tracking, it is not suitable for massive targets and unknown environments. Infrared tracking works in two ways. One is similar to RFID; the other captures infrared beacons as markers with infrared camera. The first way is inexpensive, whereas the second blocks out visual spectrum to provide clean and noise-free images for recognition. Many systems [194]–[197] use both ways for tracking. Infrared signal cannot travel through walls and easily interfere with fluorescent light and direct sunlight. Bluetooth was also used in many applications [40], [123], [198]–[202]. It is resistable to interference and easier confined to limited space but has drawback of short transmission range.

4) ULTRASONIC-BASED

Ultrasonic tracking can estimate both pose and velocity information. It is able to obtain very high tracking accuracy. However, ultrasonic emitters and receivers are rarely implemented in handheld devices. They were only used in early MAR applications [203]–[205]. Ultrasonic sensors are sensitive to temperature, occlusion and ambient noise and has been replaced by other methods in recent MAR applications.

Table 5 lists several sensors in terms of characteristics related to MAR. In addition to single sensor tracking, many systems combined different sensors to improve results. In [24], [28], [40], [141], GPS and inertial methods were combined for indoor and outdoors tracking. There are other hybrid methods such as infrared beacons and inertial sensors [194], UWB and inertial sensors [121], infrared and RFID [126] and Wi-Fi and Bluetooth [202]. We list typical sensor-based tracking methods in Table 4. Several characteristics related to MAR applications are given for comparison.

B. VISION-BASED METHODS

Vision-based tracking uses feature correspondences to estimate pose information. According to features it tracks, it can be classified into marker-based and feature-based method.

1) MARKER-BASED METHOD

Marker-based method uses fiducials and markers as artificial features. Figure 3 gives several common used markers. A fiducial has predefined geometric and properties, such as shape, size and color patterns, to make them easily identifiable. Planar fiducial is popular attributed to its superior accuracy and robustness in changing lighting conditions. Several available libraries used planar fiducial for tracking, such as ARToolkit [154], ARTag [206], ARToolKitPlus [162] and OpenTracker [150]. Many MAR applications [53], [140], [143], [148], [156] use these libraries for tracking. Mohring *et al.* [118] designed a color-coded 3D paper marker to reduce computation. It could run at 12 fps on a commercial mobile phone. The method was constrained to simple tracking related to 2D position and 1D rotation. Hagbi *et al.* [207] used shape contour concavity of patterns on planar fiducial to obtain projective-invariant signatures so that shape recognition was available from different viewpoints. As it is impractical to deploy and maintain fiducial markers in an unknown or large outdoor environment, marker-based method is only suitable for indoor applications. Besides, it suffers from problems of obtrusive, monotonous and view occlusion.

2) NATURE FEATURE METHOD

Nature feature method tracks point and region features in image sequences to calculate correspondent relationships to estimate pose information. The method requires no prior information of environment. The frame-by-frame tracking helps to remove mismatches and drift errors that most sensor-based methods suffer. However, it suffers from deficiencies of image distortion, illumination variation and self-occlusion [208]. Besides, additional registration of image sequence with real world is required to get finally results. The major problem of nature feature method is expensive in terms of computational overhead, which is especially severe for MAR applications with requirement of real time performance on mobile devices. Many researches focused on performance improvement from different aspects, such as GPU acceleration, computation outsourcing and algorithm improvement. We focus on algorithm improvement in this section with other two aspects in the next section.

Many robust local descriptors including SIFT [209] and SURF [210] are introduced for nature feature tracking in MAR field. Skrypnyk and Lowe [211] presented a traditional SIFT-based implementation. SIFT features were extracted offline from reference images and then used to compute camera pose from live video. Fritz *et al.* [212] deployed an imagery database on a remote server and conducted SIFT feature matching on the server to reduce computing and memory overhead on client mobile phones. Chen *et al.* [145] also implemented the SURF tracking on remote server. The live video captured by embedded camera was streamed to server for tracking. The result was then retrieved back to client side for display. Rosten and Drummond [213] propose a high-speed corner detection (FAST) using machine learning techniques to improve speed detection. However, FAST is not very robust to the presence of noise. PCA-SIFT [214] is a Principal Components Analysis (PCA) SIFT variant that provides more robust to image deformations, and compact descriptors than baseline SIFT; this leads to significant improvements in matching accuracy and speed.

Calonder *et al.* [215] propose to use binary strings as an efficient feature point descriptor. This descriptor similarity can be evaluated using an efficient computing algorithm such as Hamming distance, and provides higher recognition rates compared to SURF. However, it is rotation and scale invariant and it is addressed in authors' future work. Wagner *et al.* [216] replaced conventional Difference of Gaussians (DoGs) with FAST corner detector, 4*4 sub-regions with 3*3 sub-regions and k-d tree with multiple Spill Trees to reduce SIFT computation. The reduced SIFT was self-contained and tracked 6DOF at 20Hz frame rate on mobile phones. Takacs *et al.* [217] identified new contents in image sequences and conducted feature extraction and database access only when new contents appeared. They further proposed a optimization scheme [144] to decrease feature matching by rejecting irrelevant candidates according to location information. Only data from nearby location cells were considered. They further developed a feature clustering and pruning strategy to eliminate redundant information. It reduced both computation and bandwidth requirements.

An entropy-based compression scheme was used to encode SURF descriptors 7 times faster. Their method run 35% faster with only half memory consumption. Chen *et al.* [218] used integral image for Haar transformation to improve SURF computational efficiency. A Gaussian filter lookup table and an efficient arctan approximation were used to reduce floating-point computation. The method achieved performance improvement of 30% speedup. Ta *et al.* [219] proposed a space-space image pyramid structure to reduce search space of interest points. It obtained real-time object recognition and tracking with SURF algorithm. In order to speed up database querying, images were organized based on spatial relationships and only related subsets were selected for matching. The method obtained 5 times speedup compared to original SURF and it took less than 0.1s on Nokia N95 mobile phones. Wagner *et al.* [220] proposed a multiple target tracking and detection method on mobile phones. It separated target detection and tracking so that target tracking run at full frame rate and created a mask to guide detector to look for new target. To further reduce calculations, they assumed that local area around positive matched key points were visible in camera images and uncovered regions were ignored during key point detection. The method was able to track 6 planar targets simultaneously at a rate of 23 fps on an Asus P565 Windows mobile phone. Wagner *et al.* [221] propose three approaches for 6DOF natural language feature tracking for MAR in real-time systems. The authors perform a performance analysis using different feature extraction approaches such as SIFT (*PhonySIFT*) and Ferns-based (*PhonyFern*, and *PathTracker* that instead of tracking-by-detection it takes advantage of the camera and scene position difference between two succesives frames to predict the feature positions.

ORB [222] solves the rotation invariant issues of BRIEF and it is resistant to noise offering better performance than SIFT and BRIEF. BRISK [223] provides a novel method for key point detection, description and matching with lower computational cost in comparison with SIFT, SURF, and BRIEF. It can be used in task with hard real-time constrains keeping the overall performance with previous mentioned methods. Li *et al.* [224] quantized a small number of random projections of SIFT features and sent them to the server, which returned meta-data corresponding to the queried image by using a nearest neighbor search in the random projection space. The method achieved retrieval accuracy up to 95% with only 2.5kB data transmission. Park *et al.* [225] optimized original feature tracking process from various aspects. Since stable features did not lose tracking until they were out of view, there was no need to extract them from each frame. Feature point number was also limited to 20 to reduce computation. A feature prediction excluded feature points that would disappear and added new features that may appear after a period of time. It achieved runtime performance of 35ms per frame, about 7 times faster than original method.

ALIEN is a feature extraction and tracking method based on local features invariant under occlusion [226]. This algorithm can be very useful in MAR applications to provide a realistic experience in situations of object occlusion, and offer better performance, in most of the situations, than its competitors [227]–[233]. Li *et al.* [234] use two LED as fiducial markers for tracking on a hand-held device, together with the inertial sensors of the device, it can provide 6DoF pose estimation. This proposed method can improve the visualization of virtual objects in the MAR systems, and improve the AR experience; as it provides tracking of the mobile device (i.e., 6DoF position). DeCAF is a deep convolutional network open source implementation for generic tasks [235]. DeCAF (precursor of *Caffe*) outperforms the baseline SURF implementation in all of authors' experiments, and opens new feature extraction possibilities for deep neural networks in the future.

Convolutional Neural Networks (CNN) are very powerful for feature extraction as recent results indicate, [236]. In this paper, authors use an existing model for object classification (*OverFeat*) to perform feature extraction tasks. The experiment results show that although *OverFeat* was originally designed to perform object classification, it is a strong competitor for other visualization task such as feature extraction against other methods: SIFT, Bag of Words (BoW), Histogram of Gradients (HOG). *Caffe* is a BSD-licensed C++ library with Python and Matlab bindings [20]. *Caffe* provides a deep neural network framework for vision, speech and multimedia large-scale or research projects. It provides a big step forward in object classification and feature extraction performance. Besides, there are current some ports to use Caffe models in the mobile environment such as Caffe Android library [237], and CNN library [238]. Dollár *et al.* [19] introduce a novel efficient schema for computing feature pyramids (*fast feature pyramid*). Finely pyramids sampling of features of an image can improve the feature detection methods at higher computation costs. The proposed algorithm can improve significantly the speed and performance of current object and feature detection methods, decreasing the computational cost of current pyramidal methods substantially. This paper [21] propose a Fast Region-based CNN for object detection. Object detection is slow in CNN and R-CNN approaches because features are extracted in real time from each region proposal and the training is an expensive task in time and space.

Fast R-CNN uses *Caffe* framework and implements several innovative methods to improve training/testing speed and accuracy. YOLO9000 [22] is a state of the art real-time object detection system. The system can detect over 9000 categories outperforming previous R-CNN, Faster R-CNN methods. YOLO9000 framework closes the big gap that exists between object detection and classification providing object detection and classification system in real-time. The authors use WordTree to merge data from different sources and train simultaneously.

C. HYBRID TRACKING METHODS

Each individual method has its advantages and limitations. A better solution is to overcome inherent limitations of individual method by combining different methods together. For instance, inertial sensors are fast and robust under drastic rapid motion. We can couple it with vision-based tracking to provide accurate priors under fast movements. Behringer *et al.* [239] proposed a hybrid tracking algorithm to estimate the optimal 3D motion vector from displacements of 2D image features. The hybrid method employed GPS and digital compass to obtain an approximate initial position and orientation. A vision tracking then calculated camera pose by predicting new features in perspective projection of environmental models. It obtained a visual tracking precision of 0.5^o and was able to worked under a maximal rotation motion of $40^o/s$.

Jiang *et al.* [240] used gyroscope to predict orientation and image line positions. The drift was compensated by a line-based vision tracking method. A heuristic control system was integrated to guarantee system robustness by reducing re-projection error to less than 5 pixels after a long-time operation. Hu and Uchimura [241] developed a parameterized model-matching (PMM) algorithm to fuse data from GPS, 3D inertial gyroscope and vision tracking. Inertial sensor was used to evaluate initial motion and stabilize pose output. The method obtained convincing precise and robust results. Reitmayr and Drummond [125] combined vision tracking with gyroscope to get accurate results in real time. Vision tracker was used for localization and gyroscope for fast motion. A Kalman filter was used to fuse both measurements. Honkamaa *et al.* [242] used GPS position information to download Google Earth KML models, which were aligned to real world using camera pose estimation from feature tracking. The method strongly depended on the access to models on Google server.

Paucher and Turk [146] combined GPS, accelerator and magnetometer to estimate camera pose information. Images out of current view were discarded to reduce database search. A SURF algorithm was then used to match candidate images and live video to refine pose information. Langlotz *et al.* [147] used inertial and magnetic sensors to obtain absolute orientation, and GPS for current user position. A panorama-based visual tracking was then fused with sensor data by using a Kalman filter to improve accuracy and robustness.

Sensor-based method works in an open loop way. Tracking error can not be evaluated and used for further correction. Besides, it suffers from deficiencies of shielding, noise and interference. Vision-based method employs tracking result as feedback to correct error dynamically. It is analogous to closed loop system. However, tracking accuracy is sensitive to view occlusion, clutter and large variation in environment conditions. Besides, single feature-based method is prone to result inconsistence [96]. Hybrid tracking requires fusion [125], [147], [182], [241] of results from different sources. Which method to use depends on accuracy and granularity required for a specific application scenario. For instance, we can accept some tracking error if we annotate the rough outlines of a building, whereas more accurate result is required to pinpoint a particular window in the building.

VI. NETWORK CONNECTIVITY

We investigate wireless networks for pose tracking in previous section, but they are also widely used for communication and data transmission in MAR. As fast progresses in wireless network technologies and infrastructure investments, numerous systems were built on client-server architecture by leveraging wireless networks for communication. Mobile devices have various network interfaces and can be connected with a remote server either via a cellular network or using WiFi. Wearable devices usually do not have network interfaced that can be connected directly to the Web but they have bluetooth and can be connected to a companion device.

We can try to estimate the maximum amount of data to process per second for a video feed, *i.e.* to the maximum expected bandwidth required for the heaviest AR applications. Several studies suggest that the human eye transmits around 6 to 10 Mb/s to the brain by taking into account that *accurate data is available only for the central region of the retina (a circle whose diameter is 2 degrees in the visual field)*. However, most MAR hardware do not provide proper eye tracking system that would permit to isolate this area on video frames. Therefore, full frames have to be processed. If we consider that a smartphone's camera's field of view is between 60 to 70 degrees, a rough estimate of the amount of data to transmit is around 9 to 12 Gb/s. Of course, this estimate represents the upper limit of raw data that could be generated per second. Even though this amount can be drastically reduced (compression, selection of specific areas etc.), we can expect some applications to generate several hundreds of megabits per second, or even gigabits per second. Table 7 present the networking needs and limitations of MAR applications.

Regarding the latency requirements, although to the best of our knowledge there is no empirical academic study on the impact of delays for Augmented Reality, several studies measure latency for various offloading scenarios in MAR systems [13], [243], [244]. As a first reference, we can consider the recommended end-to-end one way delays for other real-time applications. Those values revolve around 100ms, depending on the application, going as low as 75ms for online gaming, while reaching 250ms for telemetry [245]. However, due to several problems such as the alignment of the virtual layer on the physical world, we can expect that a seamless experience would be characterized by way lower latencies. Michael Abrash, Chief Scientist at Occulus VR, even argues that augmented reality (AR) and virtual reality (VR) games should rely on latencies under 20ms, with a ?holy grail'' around 7ms in order to preserve the integrity of the virtual environment and prevent phenomenons such as motion sickness [246].

There exist several specifically designed platforms permit to perform offloading for AR operations. Most of them try to combine on-device operation with offloaded procedures. As those frameworks are often dealing with 3/4G networks, which are characterized by a high cost of use as well as abrupt changes in bandwidth and latencies, they try to circumscribe network usage to the most computation intensive operations. For instance, `CloudRidAR` [14] performs feature extraction from the video flow. Those features are then transmitted to the server instead of full pictures. More recently, Glimpse [247] even improves network efficiency by performing local tracking of objects, which allows to send only a select amount of frames to the server.

There are three major wireless networks used in MAR applications:

1) WIRELESS WIDE AREA NETWORK (WWAN)

WWAN is suitable for applications with large-scale mobility. There are massive WWAN implementations based on different technologies including 2G GSM and CDMA, 2.5G GPRS, 3G UMTS and 4G LTE. Higher generation network usually has much wider bandwidth and shorter latency than lower generation network. Many MAR applications [28], [29], [140], [144], [248]used WWAN for data transition and communications. One problem of WWAN is high cost of initial infrastructures investment. Besides, networks supplied by different providers are incompatible and users usually have to manually switch different networks. However, WWAN is still the most popular, sometimes the only, solution for MAR communication as it is the only available technology for wide public environments at present.

2) WIRELESS LOCAL AREA NETWORK (WLAN)

WLAN works in a much smaller scope but with higher bandwidth and lower latency. Wi-Fi and MIMO are two typical WLANs. WLAN has become popular and is suitable for indoor applications. Many MAR applications [24], [124], [145], [173] were built on WLAN based architecture. However, limited coverage may constrain it be used in wide public environments.

TABLE 6. A table comparing different types of wireless networks for MAR applications.

Type	Technology	Coverage	Bandwidth(bps)	Latency(ms)	Power(mw)
WWAN	GSM	~35km	60K	high	1000~2000
	CDMA	12km~50km	384K	high	200~100
	GPRS	~10km	56K~114K	high	~1000
	UMTS	1km~2km	2M	medium	~250
	LTE	5km~100 km[a]	5~50M	low	~ 500
WLAN	Wi-Fi	~90m	11M~54M	low	~100
	MIMO	~100m	300M	medium	unknown
WPAN	UWB	10m~100m	20M~1G	low	20~1000
	Bluetooth	~10m	1M~3M	medium	1~2.5
	ZigBee	~75m	20K~250K	low	1~100

[a]Depending on the frequency band and the topography the coverage radius changes. Low frequency bands are used in rural areas and have higher bandwidth but smaller coverage (5 km) while the so called macrocells have coverage radius of around 100 km and only acceptable perfomance.

TABLE 7. Networking Needs and Limitations in MAR applications.

Visual Input	Single HD Streaming	Uncompressed Video/Model
Latency	10 ms[a]	1 ~ 10 ms
Jitter	<1 ms	<1 ms
Throughput	6 MBps	1 GBps

[a]All the values of this table are based on the report of the second NSF Workshop on achieving ultra-low latencies in wireless networks. url: http://inlab.lab.asu.edu/nsf/

3) WIRELESS PERSONAL AREA NETWORK (WPAN)

WPAN is designed to interconnect devices, such as mobile phones, PDAs and computers, centered around individual workspace. There are many WPAN implementations including Bluetooth, ZigBee and UWB. Bluetooth and ZigBee are usually used for position tracking and data transmission [41], [53], [78], [174], whereas UWB is major for tracking. WPAN has many advantages including small volume, low power consumption and high bandwidth, but it is not suitable for application with wide whereabouts.

Table 6 gives comparison of several wireless networks used in MAR applications. All technologies have their drawbacks. We can leverage advantages of different networks to improve performance if multiple wireless networks overlap. However, it requires manually switching between different networks. Reference [249] proposed a wireless overlay network concept to choice the most appropriate available networks for use. It was totally transparent for applications and users to switch between different networks.

VII. DATA MANAGEMENT

Any practical MAR application requires efficient data management to acquire, organize and store large quantities of data. It is nature to design dedicated data management for specified applications, but it can not be reused and scaled for other applications. We require more flexible strategies to present and manage data source so as to make it available for different applications.

A. DATA ACQUISITION

MAR requires a dataset model of user environment which includs geometrical models and semantic description. Many applications create such model manually, whereas scaling it to a wide region is impractical. Data conversion from legacy geographic databases, such as GIS, is a convenient approach. Reitmayr and schmalstieg [173] extracted geographic information from a network of routes for pedestrians. Schmalstieg *et al.* [28] extracted footprints information of buildings from 2D GIS database. Some works constructed environmental model from architectural plans. Hollerer [24] created building structures from 2D map outlines of Columbia campus. As legacy database normally does not contain all necessary information, the method requires knowledge from other fields to complete modeling.

Field Survey with telemetry tools is also widely used to obtain environmental geometry data. Joseph *et al.* [250] developed a semi-automatic survey system by using fiducial markers to guide a robot for measurement, based on which Schmalstieg *et al.* [28] employed Leica Total Station TPS700 theodolite to scan indoor structure of buildings. The cloud point results were loaded into a view editor for manual construction of floors, walls and other elements. They further used a robot for surveying automatically. Output cloud points could be converted into DFX format using software packages, which was friendly to 3D applications. Results from measurement are prone to inaccuracy and noise. Besides, discrete cloud points require interpolation to rebuild cartographic presentation.

Many MAR browsers [92], [93] concerned location-based services augmented with web information. They used geolocation information, images and QR markers to search correlated contents through Internet. Results were retrieved and overlaid on current view on mobile phones. Recent products including Google Earth and Semapedia offer geo-referenced geometry data through community efforts, which are easy to access through Internet. Hossmann *et al.* [251] developed application to gather environmental data through users report of their current activities. The environmental data coverage explodes as users increase.

B. DATA MODELING

MAR Applications potentially do not access the same abstraction or presentation of dataset even with the same resource. Besides, it is difficult to guarantee presentation consistency if any change cannot be traced back to the original resource. High-level data model is required to decouple underlying resource from upper logic changes. A data model is a conceptual model to hide data details so as to facilitate understanding, representation and manipulation in a uniform manner.

Researchers at Vienna University of Technology proposed a 3-tier [28], [173] data model. The first tier was a database. The second tier linked database and application by translating raw data from database to specified data structure. The third tier was a container where all applications resided. The second tier decoupled data from presentation so that applications did not have to understand data details. Basic abstract types such as *ObjectType* and *SpatialObjectType* were predefined that application types could derive from. The XML object tree was interpreted in a geometrical way so that data storage and presentation were linked and consistent. As data were modeled with nodes, it may increase search computation for rendering engine when several information was not required.

Nicklas and Mitschang [153] also proposed a 3-layer model including client device layer, federation layer and server layer. The server layer stored resource for entire system. It could be geographical data, users' locations or virtual objects. A top-level object *Nexus Object* was designed, from which all objects such as sensors, spatial objects and event objects could inherite. The federation layer provided transparent data access to upper layer using a register mechanism. It decomposed query from client layer and then dispatched them to registers for information access. The federation layer guaranteed consistent presentation even if data servers supplied inconsistent data. The model separated underlying data operations from client layer, but it increased access delay due to delegation mechanism. Multiple copies of object on different servers caused data inconsistence.

Other than traditional 3-layer structure, Tonnis [159] proposed a 4-layer data model. The bottom layer is a dynamic peer-to-peer system to provide basic connectivity and communication services. The second layer supplied general MAR functions such as tracking, sensor management and environmental presentation. The third layer contained high-level functional modules composed of sub-layer components to offer application related functions for top layer that directly interacted with users and third-part systems. Virtual object were represented by object identifier *virtualObjectID* and their types, which were bound to a table data structure *object_properties* containing linking information. A data structure *object_relations* was proposed to describe object relationships. A special template was also used to store representative information. The flat and separate data model was more flexible and efficient for different application and rendering requirements.

C. DATA STORAGE

Since no global data repository exists, researchers have to build private data infrastructure for their MAR applications. Hollerer *et al.* [23], [24] constructed a relational central database to access meta-data based on client-server architecture. A data replication infrastructure was applied to distribute data to various clients. Piekarski and Thomas [155] and Piekarski and Bruce [156] implemented hierarchical database storage which stored and retrieved objects using hierarchical path names similar to a file system. It used virtual directories to create and store objects without understanding creation details. Reitmayr and Schmalstieg [114], [173] used a file system to organize application data in their early implementation. As file system is efficient for unstructured dataset whereas MAR data is usually well structured, Schmalstieg *et al.* [28], [53], [143] adopted a XML database with a XML-based query strategy, which was proven more flexible and efficient. Wagner and Schmalstieg [151] developed a middleware named Muddleware to facilitate database access. It had a server-side state machine to respond to any database changes. An independent thread was dispatched to control database server. Nicklas and Mitschang [153] advocated a multi-server infrastructure to decouple different data processes but it had problem of data inconsistence as data copies were deployed on multiple servers. Conventional database technologies were also widely used to store and access various resources in many MAR applications. In terms of massive data storage technologies, from user's point of view, the important issue is how to get to the most relevant information with the least effort and how to minimize information overload.

VIII. SYSTEM PERFORMANCE AND SUSTAINABILITY

Most MAR applications suffers from poor computing capability and limited energy supply. To develop applications for practical use we should consider issues of runtime performance and energy efficiency. From the developers' point of view, we should make design and development decisions based on careful task analysis.

A. RUNTIME PERFORMANCE

Recent years we have witnessed great efforts to improve runtime performance for mobile applications. A speedup from hardware, software and rendering improvements, ranging from a few times to hundreds of times, has been achieved during the past decade.

1) MULTICORE CPU PARALLELISM

There are many off-the-shelf multicore CPU processors available for mobile devices, such as dual-core Apple A6 CPU and quad-core ARM Cortex-A9 CPU. [315] reported that about 88% mobile phones would be equipped with multicore CPU by 2013. Multicore CPU consumes less energy than single core CPU with similar throughput because each core works at much lower clock frequency and voltage. Most MAR applications are composed of several basic tasks including camera access, pose tracking, network communication and rendering. Wagner and Schmalstieg [261] parallelized basic tasks for speedup. Since camera access was I/O bound rather than CPU bound, they run camera reading in a separate thread. Herling and Broll [307] leveraged multicore CPUs to accelerate SURF on mobile phones by treating each detected feature independently and assigning different features to different threads. Takacs *et al.* [316] separated detection and tracking into different threads for parallelization. The system run at 7∼10 fps on a 600MHz single-core CPU. Multithread technology is not popular for MAR applications as computing context is much more stringent than desktop computers. CPU and other accelerators are integrated into single processor system-on-chip (SoC) to share system bus. It requires intelligent scheme to schedule threads to share data and avoid access conflict.

2) GPU COMPUTING

Most mobile devices are now equipped with mobile graphics processing unit (GPU). There are many mobile GPUs including Qualcomm SnapDragon SoC with Adreno 225 GPU, TI OMAP5 SoC with PowerVR SGX 544 and Nvidia Tegra 3 SoC with ULP GeForce. Mobile GPU is developing toward programmable rendering pipeline. In order to facilitate mobile GPU programming, Khronos Group proposed a low-level graphics programming interface named OpenGL ES [252]. It supports per vertex and per pixel operations by using vertex and pixel (or fragment) shaders, which are C-like program code snippets that run on GPUs. The programmability and inherent high parallel architecture awake research interests in general computing acceleration beyond graphics rendering, namely general-purpose computing on GPU (GPGPU). However, it is complicated to program shaders as we have to map algorithms and data structure to graphics operations and data types. To alleviate the problem, Khronos Group released high-level APIs named OpenCL [253]. They also released the embedded profile of OpenCL for mobile devices.

Profile results [82], [118], [123] have shown that tracking is a major performance bottleneck for most MAR applications. Many works accelerated feature tracking and recognition with GPU on mobile devices. Wutte [254] implemented SURF on hybrid CPU and GPU with OpenGL ES 2.0. Since compression and search processes run on CPU, it required frequent data transmissions between CPU and GPU. Kayombya [255] leveraged mobile GPU to accelerate SIFT feature extraction with OpenGL ES 2.0. They broke down the process into pixel-rate sub-process and keypoint-rate sub-process and then projected them as kernels for stream processing on GPU. It took about 0.9s to 100% match keypoint positions with an image of size 200*200. Singhal *et al.* [256], [257] developed several image processing tools including Harris corner detector and SURF for handheld GPUs with OpenGL ES 2.0. They pre-computed neighboring texture coordinates in the vertex shader to avoid dependent texture reading for filtering in fragment shader. Other optimizations such as lower precision and texture compression were used to gain further improvements. Their GPU-based SURF implementation cost about 0.94s for image of size 800*480, about 1.81x speedup comparing to CPU implementation. Hofmann [258] also implemented SURF on mobile GPU with OpenGL ES 2.0. The method used mipmaps to obtain scale-aware, sub-pixel-accurate Haar wavelet sampling. It took about 0.4s to extract 1020 SURF-36 descriptors from image of size 512*384 on mobile phones. Leskela *et al.* [259] conducted several image processing tests on mobile GPU with OpenCL. The results were inspiring. However, in order to save energy consumption, mobile GPU is designed with low memory bandwidth and a few stream processors (SPs) and instruction set is also reduced.

Baek *et al.* [260] propose a parallel processing scheme using CPU an GPU for the MAR applications. The scheme processes feature extraction techniques in the CPU as it will perform better and faster; and the feature description in the GPU. The proposed present and innovative scheme that outperforms CPU only and sequential CPU-GPU schemes in AR scenarios.

3) CACHE EFFICIENCY

Many mobile devices have tiny on-chip caches around CPU and GPU to reduce latency of external memory access. For instance, Nvidia Tegra series mobile GPUs have a vertex cache for vertex fetching and a pixel and a texture caches for pixel shader. The caches are connected to a L2 cache via system bus. Memory is designed to be small to reduce energy consumption on mobile devices. Cache miss is therefore more expensive than desktop computers. Wagner and Schmalstieg [261] transmitted vertex data in interleaving way to improve cache efficiency. They further employed vertex buffer objects (VBOs) for static meshes. Many systems leverage multithreading technologies to hide memory latency. PowerVR SGX5xx GPU [262] scheduled operations from a pool of fragments when a fragment waited for texture requests. The method was effective but not energy-efficient. Besides, mobile CPU and GPU only had a few thread wraps due to power constraint, which might not hide cache misses efficiently.

Arnau *et al.* [263] decoupled pixel and texture access from fragment operations. The architecture fetched data earlier than it would be used so as to hide cache latency when it was used. It achieved 93% performance of a highly threaded implementation on desktop computer. Hofmann [258] employed mipmap technology for Haar sampling. The method obtained a beneficial side effect of significant cache efficiency as it reduced texture fetch operations from 64 to 1. Cache optimization is usually an ad hoc solution for specified problem. For instance, in [264], as problem dimension increased, data had to be stored in off-chip memory as on-chip GPU cache was not large enough. Their method was only capable for small dimensional problem. Performance improvement depends on both available hardware and problem to solve. A side benefit of cache saving is to reduce bus accesses and alleviate bandwidth traffic.

4) MEMORY BANDWIDTH SAVING

Most mobile devices adopt share storage architecture to reduce energy consumption and cost. CPU, GPU and other processors use common system bus to share memory, which makes bandwidth scarce and busy. Besides, bandwidth is designed to be small for low energy cost. Reference [265] reported that annual processor computational capability grew by 71% whereas bandwidth only by 25%, so bandwidth is more prone to be bottleneck than computation power. Date compression is usually used to alleviate the problem. Color, depth and stencil buffer data were compressed to reduce transmission on system bus [266], [267].

Texture is also compressed before it is stored in constant memory. As slight degradation of image quality is acceptable due to filter and mipmap operations during texture mapping, lossy compressions can be used to further reduce data. Moller and Strom [268] and Strom and Moller [269] proposed a hardware rasterization architecture to compress texture on mobile devices. It reduced memory bandwidth by 53%. Singhal *et al.* [257] stored texture with low precise pixel format such as RGB565, RGBA5551 and RGBA444 to reduce data occupation. Wagner and Schmalstieg [162] used native pixel format YUV12 for images captured by built-in cameras to reduce image storage.

5) RENDERING IMPROVEMENT

3D graphics rendering is one of the most computational intensive tasks on mobile device. Duguet and Drettakis [270] proposed a point-based method to reduce model presentation for cheap 3D rendering on mobile device. They represented model mesh as hierarchical points and rendered parts of them according to computational power of mobile devices. Setlur *et al.* [271] observed limitation of human perception for unimportant objects on small display screen. They augmented important objects and eliminated unimportant stuff to reduce rendering. It is an approximate rendering as some information is discarded in final result. Another approximate rendering method is programmable culling unit (PCU) proposed by Hasselgren and Moller [272]. It excluded pixel shader whose contribution to entire block of pixels is smaller than zero. The method obtained 2 times performance improvement with about 14% to 28% bandwidth saving.

It could degrade to lossy rendering if contribution factor was set to be greater than zero. Traditional immediate mode rendering (IMR) mode updates entire buffer through pipeline immediately. Mobile on-chip memory and cache are too small and rendering data has to be stored in off-chip memory. Fuchs *et al.* [273] proposed a tile-based rendering (TBR) to divide data into subsets. It performed rasterization per tile other than entire frame buffer. As tiny data was required for each tile rendering, data for a tile can be stored in on-chip memory to improve cache and bandwidth efficiency. Imagination Technologies Company delivered several TBR-based PowerVR series GPUs [262] for mobile devices. As triangles were required to sort for each tile, additional computation would increase computational cost and bandwidth usage. Performance gain depends on scene geometry structure. It obtains greater performance improvement if overdraw is high, while it may be less efficient than IMR if scene is full of long and thin triangles.

6) COMPUTATION OUTSOURCING

As rapid advances in high bandwidth, low latency and wide deployment of wireless network, it is feasible to outsource computational intensive and even entire workloads to remote server and cloud.

a: LOCAL-RENDERING REMOTE-COMPUTING (LRRC)

Computational tasks are outsourced to remote server for acceleration and results are retrieved back to client for rendering and further processes. Chang and Ger [274] proposed an image-based rendering (IBR) method to display complex models on mobile devices. They used a computational intensive ray-tracing algorithm to get depth image of geometry model on the server. The depth image contained rendering data such as view matrix, color and depth buffers, which were sent back to client mobile devices for rendering with a wrapping equation [275]. It could run about 6 fps on a 206MHz StrongArm mobile processor. Fritz *et al.* [212] conducted object detection and recognition with a modified SIFT algorithm on server to search database for object information descriptor, which was sent back to client for annotation. It run 8 times faster with 98% accuracy than original implementation. Chen *et al.* [145] streamed live videos on mobile phone to remote server, on which a SURF-based recognition engine was used to obtain features. It took only 1.0 second recognition latency to detect 150 SURF features from each 320*240 frame. Gu *et al.* [276] conducted marker detection on server and processed graphics rendering on mobile client. Captured live video was compressed with YUV420 format to reduce data transmission.

b: REMOTE-RENDERING REMOTE-COMPUTING (R3C)

computational and rendering tasks are both outsourced to remote servers and client mobile devices are used as display and user interface. Pasman and Woodward [123] imported the ARToolkit onto remote server for marker tracking. Virtual overlays were blended with captured images on server side, which were encoded and sent back to client for display. It took about 0.8s to visualize a 3D model with 60,000 polygons on a PDA ten years ago. Lamberti *et al.* [277] visualized 3D scene on remote clusters by using Chromium system to split computational task. Results were reassembled and sent back to client PDA as still images stream. It could display complex models realistically at interactive frame rate. They [278] further transmit video stream to server. The server could tailor results according to screen resolution and bandwidth to guarantee realtime performance.

With fast development of wireless network and cloud computing technologies, a few works imported computational and rendering tasks into cloud to gain performance improvement. Luo [279] proposed conception of Cloud-Mobile Convergence for Virtual Reality (CMCVR) to improve VR/AR system based on cloud-based infrastructure. He implemented a complicated vision-based gesture interface based on original system. Runtime performance was almost not impacted with help of cloud computing. Lu *et al.* [280] outsourced whole task on cloud. User input was projected to cloud and rendered screen was compressed and sent back to mobile devices for display.

In R3C, as only rendered images are required for display on client mobile devices, transmitted data amount is irrelevant to complexity of virtual scene. In addition, it is free of application migration and compatible problems because all tasks are completed on server and cloud. Both LRRC and R3C suffer from several common problems. The performance is limited by network characteristics including shielding, bandwidth and latency. Data compression alleviates bandwidth traffic to a certain extent at the price of additional computations. In [281] and [282], several principles in terms of bandwidth and computational workload were proposed to guide usage of computation offloading. Comparing to local computation, remote computation also has privacy security problem.

c: APPROXIMATE COMPUTING

The first ingredient of approximate computing is a system that can trade off accuracy for efficiency. Xu *et al.* [283], Mittal [284], and Han and Orshansky [285] describe the reasons behind approximate computing and explore the current approximation techniques in hardware and software. This trade off can bring real-time speeds in very complex computational task such as image processing and sensing (i.e., GPS location, motion sensors). Moreau *et al.* [286] propose a hardware approximate computing approach in form of a flexible FPGA-based neural network for approximate programs. Their hardware approach demonstrates higher speeds and better energy savings for current applications that use neural networks as accelerators. In [287], it describes a framework to facilitate application resilient characterization (ARC) to bring approximate computing closer to mainstream adoption so future adopters of this technique can analyze and characterize the resilience of their applications. Vassiliadis *et al.* [288] propose a framework for energy-constrained execution with controlled and graceful quality loss. A simple programming model allows developers to structure the computation in different tasks, and to express the relative importance of these tasks for the quality of the end result. Furthermore, Sampson *et al.* [289] describe ACCEPT, a compiler framework for approximation that balances automation with programming guidance. Authors demonstrate that ACCEPT can improve end-to-end performance with very low quality degradation. Although, approximate computing is still in early steps, MAR applications that can tolerate imprecision (i.e., image recognition, motion sensing, location) can be far more efficient, in terms of energy saving and speed, when we let them operate imprecisely.

B. ENERGY EFFICIENCY

Survey [290] has revealed that battery life is one of the most studied problem for handheld users. It is especially emphasized for MAR application due to their power-hungry nature. Energy consumption can be addressed at various levels from hardware platform, sensor, network and tracking algorithms to user interaction. Rodriguez and Crowcroft [291] gave a detailed survey on energy management for mobile devices. Most methods mentioned are also suitable for MAR applications. In this section we investigate several power saving technologies not mentioned in the survey, but may be useful for MAR applications. Semiconductor principle proves that power consumption is exponential to frequency and voltage. Single core CPU improves computational performance with an exponential jump in power consumption, whereas multicore CPU improves performance at the cost of linear increment in power consumption attributed to the fact that each core run at low frequency and voltage when workload is allocated to multiple cores. Nvidia [292] showed a 40% power improvement by using multicore low-frequency CPU. High parallelism at low clock frequency is much more energy efficient than low parallelism at high clock frequency. MAR applications can also benefits energy saving from leveraging multicore CPU for performance acceleration .

Energy consumption is also proportional to memory access. It is an efficient way to reduce energy consumption by limiting memory access and bandwidth traffic. Arnau *et al.* [263] decoupled memory access from fragment calculation to obtain 34% energy saving. Bandwidth saving technologies, such as PCU [272], TBR [262] and data compression [266], [267], also reduce energy consumption as long as energy reduction is greater than energy exhausted by additional operations. Sohn *et al.* [293] designed a low-power 3D mobile graphics processor by dynamic configuration of memory according to bandwidth requirements. It reduced power consumption by partly activation of local memory to meet 3D rendering operations. We can save energy by putting task and hardware components that do not work to sleep . Clock gating [266] is widely used in the design of mobile devices for energy saving from circuit level. Mochocki *et al.* [294] employed dynamic voltage and frequency scaling (DVFS) technology to optimize mobile 3D rendering pipeline based on previous analytical results. The method was able to reduce power consumption by 40%. For client-server based applications, the network interfaces are idle in most time to wait for data. It was scheduled to work asynchronously to interleave with other tasks so as to save energy consumption [261].

Computation offloading also conditionally improves energy efficiency. Kumar and Lu [295] proposed a model to evaluate energy consumption by using computation offloading methods. It indicated that energy improvement depend on computation, wireless bandwidth and data amount to transmit through network. System also benefits from energy efficiency if a large amount of computation is offloaded with limited data to transmit. Kosta *et al.* [296] proposed a ThinkAir framework to help developers migrate mobile applications to the cloud. The framework provided method-level computation offloading. It used multiple virtual machine (VM) images to parallelize method execution. Furthermore apart from the assistance to be provided by remote cloud servers [296]–[299] it can also be provided by nearby cloudlets [300]–[303], or even by nearby mobile devices [304]–[306]. In all these cases, if it is properly done the computation offloading improves the performance of the application while spends less power. Since MAR is typical computationally intensive and current network bandwidth is relative high, most MAR applications obtain energy benefit from computation offloading.

C. PERFORMANCE VS. ENERGY EFFICIENCY

As mobile processors are usually designed with emphasis on lower power consumption rather than performance [264], energy efficiency is even more important than performance on mobile devices. In certain conditions, performance promotion also improves energy efficiency, whereas in other cases it increases energy consumption. If we improve performance with operation reduction such as decreasing data amount and memory access [262], [266], [267], it will also decrease power consumption; if we improve performance by using more powerful hardware [258], [307], it will increase energy consumption. In most cases ''increasing'' method obtain higher performance gain than ''decreasing'' way with the cost of higher energy consumption. A more practical solution is to combine them by fully using available computing resource with the consideration of energy saving.

IX. CHALLENGING PROBLEMS

The continuously increasing needs of the MAR application users increase the competition between application developers to build more attractive, fancier and more innovative applications. The technological advancements can not follow this pace and this forces mobile developers to adapt solutions such as computation outsourcing, as we presented in the previous section. However, there exist various other obstacles, such as the battery capabilities of smart devices and communication delay between them and the cloud servers, that the application developers need to wait for the proper technology to be available and affordable. We discuss such limitations and we comment on the uniqueness and the innovativeness of the MAR applications in Section IX-A. Apart from the technical challenges in MAR, there exist various concerns regarding the security and the privacy of such applications. Section IX-B contains more details about these and Section IX-C discuss the social acceptance of MAR devices from people.

A. TECHNOLOGICAL LIMITATIONS

MAR developments are mostly based on various technologies as mentioned above. Many problems such as network QoS deficiency and display limitations remain unsolved in their own fields. Some problems are induced by the combination of multiple technologies. Below we discuss the open issues in various technological sections.

1) ENERGY EFFICIENCY

The batteries of mobile devices are designed to be sustainable for common functions such as picture capturing and Internet access, which are supposed to be used at intervals. MAR applications require long-time cooperation of cameras capturing, GPS receiving and Internet connection. These tasks, by being energy hungry even when they are working alone, when working together drain the battery very quickly. So, in order for MAR applications to be able to be deployed in common mobile devices, an improvement in the common batteries is needed. Approximate Computing can deal with energy hungry MAR applications by reducing the accuracy of heavy computations but this process, if not properly conducted, can compromise the performance of the application and the quality of UX.

2) LOW-LEVEL MAR LIBRARIES

Although there is a fast progress in the areas of computation offloading, cooperative computing and more generally on network assisted technologies, many MAR systems are designed to be self-contained to make it free from environmental support. Self-support is emphasized to map completely unknown surroundings and improve user experience. However, this decision introduces complexity and limitations. For instance, many systems employ visual feature method to get rid of deployed visual markers, but deficiencies of heavy computational overhead and poor robustness make the system even less applicable for most applications. Following this monolithic apporach, most MAR application developers reimplement the basic functionalities that are required by their applications. Besides, what useful annotations can be expected if we know nothing about the environment? It is still unclear about the necessity to make it universal for completely unprepared surroundings. With the development of pervasive computing and Internet of Things (IoT), computing and identification are woven into the fabric of daily life and indistinguishable from environments. Such systems should be deeply integrated with environment other than isolated from it. Another reason to emphasize self-support is outdoor usage. A study case showed that GPS usage coverage was much lower than expected [308]. As GPS was shielded in indoor environments, it indicated that users may spent most of their time indoors, so there may be not so great urgency to make system completely self-contained. So, in order for MAR applications to be able to be more deployable and assist application developers to produce functional MAR applications, we argue that system level support for the basic functionalities, like object tracking, positioning, computation offloading and others, is needed.

3) KILLER MAR APPLICATIONS

Most existing MAR applications are only prototypes for experiment and demonstration purposes. MAR has great potential to change our ways to interact with real world, but it still lacks killer applications to show its capabilities, which may make it less attractive for most users. Breakthrough applications are more likely to provide a way for drivers to see through buildings to avoid cars coming from cross streets or help backhoe operator to watch out fiber-optic cables buried underground in the field. We have witness similar experience for Virtual Reality (VR) development during past decades, so we should create feasible applications to avoid risk of the same damage to MAR as seen when VR was turned from hype to oblivion. Smart glasses is a milestone device to raise public interest but it is still stuck with absence of killer applications.

4) NETWORKING

Another crucial problem is the network requirements due to the computation offloading and the remote access to cloud databases for virtual objects. The requirement for high user quality of experience, that depends on the frame rate of the MAR application, implies that the remotely executed parts of the application have be processed in a remote server with which the mobile user has low communication delay. Unfortunately, this implies the deployment of edge computing clusters, which have a high monetary cost. The soon-to-be-available 5G together with recent research works have provision regarding the experience improvement of the so-called *killer applications* for MAR [309]–[311]. Furthermore, many high accurate tracking approaches are available in computer vision fields but they can not be directly used on mobile devices due to limited computing capability. We have discussed technology related challenges in previous sections. Several papers [4], [8], [24] also made detailed investigation of it.

5) DATA MANAGEMENT

Apart from the networking issues, there exist a huge space for improvement in virtual object management. Most MAR applications have pre-stored all the virtual objects that can potentially visualise in the screen but this tactic, apart from increasing the needs of the application, imposes serious constraints in the potential functionalities of it. We argue that a MAR killer application should get the virtual objects from cloud databases using efficient caching and prefetching techniques. In this way, sky is the limit in the number and categories of them. Also, publicly available virtual object cloud databases can be shared by multiple applications and allow application developers to focus only on the functionalities of the applications and not on the object collection and their registration and association with other characteristics like location.

6) UI/UX

The design of user friendly UI, and empowering UX for MAR applications are important aspects that developers and researchers need to address in their MAR approaches. Most of the previous work mention the necessity of some guidelines to achieve a clear interface that enhance users to interact with the AR content. Furthermore, future work has to focus in collaborative environments where users can share and interact with other users. Researchers need to find better ways to personalize the current AR content to enhance the UX. The interaction with AR objects does not always need to be an analogy of real-world interactions (i.e., hand rotation to rotate objects), as some participants suggested other faster methods to interact with them (i.e., move finger up and down to rotate the AR object). The interaction user-AR content has to be fluent and give accurate feedback in order to maintain good user experience. Furthermore, MAR applications need to provide a useful experience with relevant information, that improve the flow experience.

MAR also has several intrinsic problems from technology aspect, which are very much underexplored but worth great consideration. We address these challenges in the following sections.

B. SECURITY AND PRIVACY

Privacy and security are especially important for MAR due to various potential invasion sources including personal identification, location tracking and private data storage. Many MAR applications depend on personal location information to provide services. For instance, in client-server applications, user's position is transmitted to third-party server for tracking and analysis, which may be collected over time to trace user activity. It is more serious for collaborative MAR applications as users have to share information with others, which not only provides opportunity for others to snoop around private information but also raise concern of how to trust quality and authenticity of user-generated information supplied by others. *ReadMe* [17], for example, is an innovative real-time recommendation system for MAR ecosystems. The role of *ReadMe* is to provide the adequate object to render on the user's screen based on the user's context. The systems analyse the user's characteristics and the dynamic features of the mobile environment in order to decide which objects need to be rendered. However, the authors or *ReadMe* do not consider the users' concerns about the required data that make the system functional and the recommendations suitable. To guarantee privacy safety, we require both trust models for data generation and certification mechanisms for data access. But the generated data from such systems are usually not stored locally and this fact raises privacy concerns to the users. Google Glass is a popular example to show users' concern about privacy. Although it is not widely delivered, it has already raised privacy concern that users can identify strangers by using facial recognition or surreptitiously record and broadcast private conversations.

Acquisti *et al.* [312] discussed privacy problem of facial recognition for AR applications. Their experiment implied that users could obtain inferable sensitive information by face matching against facial images from online sources such as social network sites (SNS). They proposed several privacy guidelines including openness, use limitation, purpose specification and collection limitation to protect privacy use. There is a big concern about privacy in Big Data and AR ecosystems [101]. The privacy concern is a serious issue nowadays, although individual data is fragmented or even hashed there are already studies that suggest that is possible to correlate individual patterns from Big Data sources of information. AR and MAR applications should follow some guidelines (i.e., forceful laws and regulation) to avoid privacy leakage or malicious purposes. In the context of Visual Privacy, Shu *et al.* [313] implemented Cardea, a context aware visual privacy protection framework from pervasive cameras. Cardea allows mobile users to avoid having their photo taken by non familiar people while allows hand gestures that give flexibility to the users. Furthermore, the same authors implemented an updated version of Cardea that allows both hand gestures and visual tags [314].

C. SOCIAL ACCEPTANCE

Many factors such as device intrusion, privacy and safety considerations may affect social acceptance of MAR. To reduce system intrusion, we should both miniaturize computing and display devices and supply a nature interactive interface. Early applications equipped with backpack laptop computer and HMDs introduce serious device intrusion. Progress in miniaturization and performance of mobile devices alleviate the problem to certain extent, but they do not work in a nature way. Users have to raise their hands to point cameras at real objects during system operation, which may cause physical fatigue for users. The privacy problem is also seriously concerned. A ''Stop the Cyborgs'' movement has attempted to convince people to ban Google Glass in their premises. Many companies also post anti-Google Glass signs in their offices. As MAR distracts users' attention from real world, it also induce safety problems when users are operating motor vehicles or walking in the streets. All these issues work together to hurdle the social acceptance of MAR technology.

X. CONCLUSION

In this paper, we provide a complete and detailed survey of the advances in mobile augmented reality in terms of application fields, user interfaces and experience metrics, system components, object tracking and registration, network connectivity and data management, system performance and sustainability and we conclude with challenging problems. Although there are still several problems from technical and application aspects, it is estimated one of the most promising mobile applications. MAR has become an important manner to interact with real world and will change our daily life.

The booming development in cloud computing and wireless networks, mobile cloud computing becomes a new trend to combine the high flexibility of mobile devices and the high-performance capabilities of cloud computing. It will play a key role in MAR applications since it can undertake heavy computational tasks to save energy and extend battery lifetime. Furthermore, cloud services for MAR applications can operate as caches and decrease the computational cost for both MAR devices and cloud providers. As MAR applications run on a remote server, we can also overcome limitations of mobile operating systems with help of mobile browsers. It is possible to combines multiple mobile devices for cooperative mobile computing which will be suitable for collaborative MAR applications such as multi-player games, collaborative design and virtual meeting. Although there are still several problems such as bandwidth limitation, service availability, heterogeneity and security, mobile cloud computing and cooperative mobile computing seem promising new technologies to promote MAR development to a higher level. Present MAR applications are mostly limited to mobile phones. We believe that these mobile devices are transient choices for MAR as they are not originally designed for MAR purpose. They happen to be suitable but not perfect for it. Only dedicated devices such as Google Glass, Microsoft Hololens and Madgaze can fully explore potential capability of MAR but in their current state they luck of resources. As the development of mobile computing and wearable computers such as AR glass and wristwatch devices is keeping pace, we look forward to its renaissance.

REFERENCES

[1] Juniper Research. (2009). ''Augmented reality on the mobile to generate 732 million dollars by 2014.'' [Online]. Available: https://juniperresearch.com/viewpressrelease.php?pr=166

[2] P. Renevier and L. Nigay, ''Mobile collaborative augmented reality: The augmented stroll,'' in *Proc. 8th IFIP Int. Conf. Eng. Human-Comput. Interaction (EHCI)*, London, U.K., 2001, pp. 299–316. [Online]. Available: http://dl.acm.org/citation.cfm?id=645350.650720

[3] T. Kock, ''The future directions of mobile augmented reality applications,'' in *Proc. PIXEL*, 2010, pp. 1–10.

[4] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, ''Recent advances in augmented reality,'' *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, Nov./Dec. 2001.

[5] R. T. Azuma, ''A survey of augmented reality,'' *Presence, Teleoper. Virtual Environ.*, vol. 6, no. 4, pp. 355–385, Aug. 1997. [Online]. Available: http://dx.doi.org/10.1162/pres.1997.6.4.355

[6] H. A. Karimi and A. Hammad, Eds., *Telegeoinformatics: Location-Based Computing and Services*. Boca Raton, FL, USA: CRC Press, 2004.

[7] M. A. Schnabel, X. Wang, H. Seichter, and T. Kvan, ''From virtuality to reality and back,'' *Proc. Int. Assoc. Soc. Design Res.*, vol. 1, p. 15, Nov. 2007.

[8] G. Papagiannakis, G. Singh, and N. Magnenat-Thalmann, ''A survey of mobile and wireless technologies for augmented reality systems,'' *Comput. Animation Virtual Worlds*, vol. 19, no. 1, pp. 3–22, 2008.

[9] M. Billinghurst, A. Clark, and G. Lee, ''A survey of augmented reality,'' *Found. Trends Human-Comput. Interact.*, vol. 8, nos. 2–3, pp. 73–272, 2015. [Online]. Available: http://dx.doi.org/10.1561/1100000049

[10] P. Geiger, M. Schickler, R. Pryss, J. Schobel, and M. Reichert, ''Location-based mobile augmented reality applications: Challenges, examples, lessons learned,'' in *Proc. 10th Int. Conf. Web Inf. Syst. Technol. (WEBIST)*, 2014, pp. 383–394.

[11] B. Shi, J. Yang, Z. Huang, and P. Hui, ''Offloading guidelines for augmented reality applications on wearable devices,'' in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2015, pp. 1271–1274. [Online]. Available: http://doi.acm.org/10.1145/2733373.2806402

[12] T. Schürg, ''Development and evaluation of interaction concepts for mobile augmented and virtual reality applications considering external controllers,'' Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, Germany, Tech. Rep. 52062, 2015.

[13] P. Jain, J. Manweiler, and R. R. Choudhury, ''OverLay: Practical mobile augmented reality,'' in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2015, pp. 331–344.

[14] Z. Huang, W. Li, P. Hui, and C. Peylo, ''CloudRidAR: A cloud-based architecture for mobile augmented reality,'' in *Proc. Workshop Mobile Augmented Reality Robotic Technol.-Based Syst. (MARS)*, New York, NY, USA, 2014, pp. 29–34. [Online]. Available: http://doi.acm.org/10.1145/2609829.2609832

[15] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, "Real-time computer vision with OpenCV," *Commun. ACM*, vol. 55, no. 6, pp. 61–69, Jun. 2012.
[16] B.-K. Seo, J. Park, H. Park, and J.-I. Park, "Real-time visual tracking of less textured three-dimensional objects on mobile platforms," *Opt. Eng.*, vol. 51, no. 12, p. 127202, 2013. [Online]. Available: http://dx.doi.org/10.1117/1.OE.51.12.127202
[17] D. Chatzopoulos and P. Hui, "ReadMe: A real-time recommendation system for mobile augmented reality ecosystems," in *Proc. ACM Multimedia Conf. (MM)*, New York, NY, USA, 2016, pp. 312–316. [Online]. Available: http://doi.acm.org/10.1145/2964284.2967233
[18] C. Curino *et al.*, "Relational cloud: A database-as-a-service for the cloud," in *Proc. 5th Biennial Conf. Innov. Data Syst. Res.*, Asilomar, CA, USA, Jan. 2011.
[19] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.
[20] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[22] R. Girshick, "Fast R-CNN," in *Proc. 2015 IEEE Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.169
[23] T. Höllerer *et al.*, "User interface management techniques for collaborative mobile augmented reality," *Comput. Graph.*, vol. 25, no. 5, pp. 799–810, 2001.
[24] T. H. Hollerer, "User interfaces for mobile augmented reality systems," Ph.D. dissertation, Dept. Comput. Sci., Columbia Univ., New York, NY, USA, 2004.
[25] P. Dahne and J. N. Karigiannis, "Archeoguide: System architecture of a mobile outdoor augmented reality system," in *Proc. ISMAR*, 2002, pp. 263–264.
[26] P. Fockler, T. Zeidler, B. Brombach, E. Bruns, and O. Bimber, "PhoneGuide: Museum guidance supported by on-device object recognition on mobile phones," in *Proc. Mobile Ubiquitous Multimedia*, 2005, pp. 3–10.
[27] N. Elmqvist, D. Axblom, and J. Claesson, "3DVN: A mixed reality platform for mobile navigation assistance," in *Proc. CHI*, 2006, pp. 1–4.
[28] D. Schmalstieg *et al.*, "Managing complex augmented reality models," *IEEE Comput. Graph. Appl.*, vol. 27, no. 4, pp. 48–57, Jul. 2007.
[29] Y. Tokusho and S. Feiner, "Prototyping an outdoor mobile augmented reality street view application," in *Proc. ISMAR*, 2009, pp. 3–5.
[30] R. M. De and D. Dasgupta, "Mobile tourism planning," *IBM Acad. TechNotes*, vol. 3, no. 5, 2012. [Online]. Available: http://docshare01.docshare.tips/files/26847/268472933.pdf
[31] S. Vert and R. Vasiu, "Relevant aspects for the integration of linked data in mobile augmented reality applications for tourism," in *Proc. Int. Conf. Inf. Softw. Technol.*, 2014, pp. 334–345.
[32] V. Kasapakis, D. Gavalas, and P. Galatis, "Augmented reality in cultural heritage: Field of view awareness in an archaeological site mobile guide," *J. Ambient Intell. Smart Environ.*, vol. 8, no. 5, pp. 501–514, 2016.
[33] A. A. M. Pires, "Combining paper maps and smartphones in the exploration of cultural heritage," Ph.D. dissertation, Dept. Faculty Sci. Technol., Univ. Nova Lisboa, Lisbon, Portugal, 2016.
[34] W. Narzt *et al.*, "Augmented reality navigation systems," *Universal Access Inf. Soc.*, vol. 4, no. 3, pp. 177–187, 2006.
[35] F. Wedyan, R. Freihat, I. Aloqily, and S. Wedyan, "JoGuide: A mobile augmented reality application for locating and describing surrounding sites," in *Proc. 9th Int. Conf. Adv. Comput.-Human Interactions*, Venice, Italy, Apr. 2016.
[36] Z. Shi, H. Wang, W. Wei, X. Zheng, M. Zhao, and J. Zhao, "A novel individual location recommendation system based on mobile augmented reality," in *Proc. IEEE Int. Conf. Identificat., Inf., Knowl. Internet Things (IIKI)*, Oct. 2015, pp. 215–218.
[37] S. Vert and R. Vasiu, "Integrating linked open data in mobile augmented reality applications—A case study," *TEM J.*, vol. 4, no. 1, pp. 35–43, 2015.
[38] A. H. Behzadan, B. W. Timm, and V. R. Kamat, "General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering," *Adv. Eng. Inform.*, vol. 22, no. 1, pp. 90–105, 2008.
[39] W. Piekarski and B. Thomas, "ARQuake: The outdoor augmented reality gaming system," *Commun. ACM*, vol. 45, no. 1, pp. 36–38, 2002.
[40] A. D. Cheok *et al.*, "Human Pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area," in *Proc. Mobile HCI*, 2003, pp. 209–224.
[41] A. Henrysson, M. Billinghurst, and M. Ollila, "Face to face collaborative AR on mobile phones," in *Proc. ISMAR*, 2005, pp. 80–89.
[42] M. Hakkarainen and C. Woodward, "SymBall: Camera driven table tennis for mobile phones," in *Proc. ACE*, 2005, pp. 391–392.
[43] A. Morrison *et al.*, "Like bees around the hive: A comparative study of a mobile augmented reality map," in *Proc. CHI*, 2009, pp. 1889–1898.
[44] Business Insider. (2012). "Amazing augmented reality ads." [Online]. Available: http://www.businessinsider.com/11-amazing-augmented-reality-ads-2012-1
[45] A. Panayiotou, "RGB Slemmings: An augmented reality game in your room," M.S. thesis, Org. Utrecht Univ., Utrecht, The Netherlands, 2016.
[46] D. Chalvatzaras, N. Yiannoutsou, C. Sintoris, and N. Avouris, "Do you remember that building? Exploring old Zakynthos through an augmented reality mobile game," in *Proc. Int. Conf. Interactive Mobile Commun. Technol. Learn. (IMCL)*, Nov. 2014, pp. 222–225.
[47] Y.-G. Kim and W.-J. Kim, "Implementation of augmented reality system for smartphone advertisements," *Int. J. Multimedia Ubiquitous Eng.*, vol. 9, no. 2, pp. 385–392, 2014.
[48] T. Olsson, E. Lagerstam, T. Kärkkäinen, and K. Väänänen-Vainio-Mattila, "Expected user experience of mobile augmented reality services: A user study in the context of shopping centres," *Pers. Ubiquitous Comput.*, vol. 17, no. 2, pp. 287–304, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-011-0494-x
[49] S. G. Dacko, "Enabling smart retail settings via mobile augmented reality shopping apps," *Technol. Forecasting Social Change*, to be published. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040162516304243
[50] S. J. Yohan, S. Julier, Y. Baillot, M. Lanzagorta, D. Brown, and L. Rosenblum, "Bars: Battlefield augmented reality system," in *Proc. NATO Inf. Syst. Technol. Panel Symp. New Inf. Process. Techn. Military Syst.*, 2000, pp. 9–11.
[51] M. Träskbäack and H. Haller, "Mixed reality training application for an oil refinery: User requirements," in *Proc. VRCAI*, 2004, pp. 324–327.
[52] E. Klopfer, J. Perry, K. Squire, M.-F. Jan, and C. Steinkuehler, "Mystery at the museum: A collaborative game for museum education," in *Proc. CSCL*, 2005, pp. 316–320.
[53] D. Schmalstieg and D. Wagner, "Experiences with handheld augmented reality," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 3–18.
[54] R. Freitas and P. Campos, "SMART: A system of augmented reality for teaching 2nd grade students," in *Proc. Brit. Comput. Soc. Conf. Human-Comput. Interact.*, 2008, pp. 27–30.
[55] Á. Di Serio, M. B. Ibáñez, and C. D. Kloos, "Impact of an augmented reality system on students' motivation for a visual art course," *Comput. Edu.*, vol. 68, pp. 586–596, Oct. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131512000590
[56] M. Bower, C. Howe, N. McCredie, A. Robinson, and D. Grover, "Augmented reality in education—Cases, places and potentials," *Edu. Media Int.*, vol. 51, no. 1, pp. 1–15, 2014.
[57] G. Koutromanos, A. Sofos, and L. Avraamidou, "The use of augmented reality games in education: A review of the literature," *Edu. Media Int.*, vol. 52, no. 4, pp. 253–271, 2015. [Online]. Available: http://dx.doi.org/10.1080/09523987.2015.1125988
[58] P. Chen, X. Liu, W. Cheng, and R. Huang, "A review of using augmented reality in education from 2011 to 2016," in *Innovations in Smart Learning*. Springer, 2017, pp. 13–18.
[59] H. E. Pence, "Smartphones, smart objects, and augmented reality," *Ref. Librarian*, vol. 52, nos. 1–2, pp. 136–145, 2010. [Online]. Available: http://dx.doi.org/10.1080/02763877.2011.528281
[60] C.-H. Lan, S. Chao, Kinshuk, and K.-H. Chao, "Mobile augmented reality in supporting peer assessment: An implementation in a fundamental design course," presented at the Int. Assoc. Develop. Inform. Soc. (IADIS), Oct. 2013. [Online]. Available: https://eric.ed.gov/?id=ED562244
[61] L. P. Prieto, Y. Wen, D. Caballero, and P. Dillenbourg, "Review of augmented paper systems in education: An orchestration perspective," *J. Edu. Technol. Soc.*, vol. 17, no. 4, pp. 169–185, 2014. [Online]. Available: http://www.jstor.org/stable/jeductechsoci.17.4.169

[62] Y. Shuo, M. Kim, E.-J. Choi, and S. Kim, ''A design of augmented reality system based on real-world illumination environment for edutainment,'' *Indian J. Sci. Technol.*, vol. 8, no. 25, pp. 1–6, 2015.
[63] E. FitzGerald, R. Ferguson, A. Adams, M. Gaved, Y. Mor, and R. Thomas, ''Augmented reality and mobile learning: The state of the art,'' *Int. J. Mobile Blended Learn.*, vol. 5, no. 4, pp. 43–58, 2013.
[64] T. H. C. Chiang, S. J. H. Yang, and G.-J. Hwang, ''An augmented reality-based mobile learning system to improve students' learning achievements and motivations in natural science inquiry activities,'' *J. Edu. Technol. Soc.*, vol. 17, no. 4, pp. 352–365, 2014.
[65] M. Dunleavy and C. Dede, ''Augmented reality teaching and learning,'' in *Handbook of Research on Educational Communications and Technology*. New York, NY, USA: Springer, 2014, pp. 735–745.
[66] H.-Y. Chang, H.-K. Wu, and Y.-S. Hsu, ''Integrating a mobile augmented reality activity to contextualize student learning of a socioscientific issue,'' *Brit. J. Edu. Technol.*, vol. 44, no. 3, pp. E95–E99, 2013.
[67] J. Bacca, S. Baldiris, R. Fabregat, S. Graf, and Kinshuk, ''Augmented reality trends in education: A systematic review of research and applications,'' *J. Edu. Technol. Soc.*, vol. 17, no. 4, pp. 133–149, 2014. [Online]. Available: http://www.kinshuk.info/
[68] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, ''Current status, opportunities and challenges of augmented reality in education,'' *Comput. Edu.*, vol. 62, pp. 41–49, Mar. 2013.
[69] C. Kamphuis, E. Barsom, M. Schijven, and N. Christoph, ''Augmented reality in medical education?'' *Perspect. Med. Edu.*, vol. 3, no. 4, pp. 300–311, 2014.
[70] M. Billinghurst and A. Duenser, ''Augmented reality in the classroom,'' *Computer*, vol. 45, no. 7, pp. 56–63, 2012.
[71] H. Kaufmann, ''Construct3D: An augmented reality application for mathematics and geometry education,'' in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 656–657.
[72] Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz, '''Studierstube': An environment for collaboration in augmented reality,'' *Virtual Reality*, vol. 3, no. 1, pp. 37–48, 1998.
[73] K.-E. Chang, C.-T. Chang, H.-T. Hou, Y.-T. Sung, H.-L. Chao, and C.-M. Lee, ''Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum,'' *Comput. Edu.*, vol. 71, pp. 185–197, Feb. 2014.
[74] I. Radu, ''Augmented reality in education: A meta-review and cross-media analysis,'' *Pers. Ubiquitous Comput.*, vol. 18, no. 6, pp. 1533–1543, 2014.
[75] Y. Baillot, D. Brown, and S. Julier, ''Authoring of physical models using mobile computers,'' in *Proc. ISWC*, 2001, pp. 39–46.
[76] W. Piekarski and B. H. Thomas, ''Interactive augmented reality techniques for construction at a distance of 3D geometry,'' in *Proc. Eurograph. Virtual Environ.*, 2003, pp. 19–28.
[77] F. Ledermann and D. Schmalstieg, ''APRIL: A high-level framework for creating augmented reality presentations,'' in *Proc. IEEE Virtual Reality*, Mar. 2005, pp. 187–194.
[78] A. Henrysson, M. Ollila, and M. Billinghurst, ''Mobile phone based AR scene assembly,'' in *Proc. MUM*, 2005, pp. 95–102.
[79] O. Bergig, N. Hagbi, J. El-Sana, and M. Billinghurst, ''In-place 3D sketching for authoring and augmenting mechanical systems,'' in *Proc. ISMAR*, 2009, pp. 87–94.
[80] N. Hagbi, R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana, ''In-place sketching for content authoring in augmented reality games,'' in *Proc. VR*, 2010, pp. 91–94.
[81] G. Klinker, O. Creighton, A. H. Dutoit, R. Kobylinski, C. Vilsmeier, and B. Brugge, ''Augmented maintenance of powerplants: A prototyping case study of a mobile AR system,'' in *Proc. ISMAR*, 2001, pp. 214–233.
[82] M. Billinghurst, M. Hakkarainen, and C. Wodward, ''Augmented assembly using a mobile phone,'' in *Proc. 7th Int. Conf. Mobile Ubiquitous Multimedia*, 2008, pp. 84–87.
[83] S. J. Henderson and S. K. Feiner, ''Augmented reality in the psychomotor phase of a procedural task,'' in *Proc. ISMAR*, 2011, pp. 191–200.
[84] A. Tang, C. Owen, F. Biocca, and W. Mou, ''Comparative effectiveness of augmented reality in object assembly,'' in *Proc. CHI*, 2003, pp. 73–80.
[85] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, ''An augmented reality training platform for assembly and maintenance skills,'' *Robot. Auto. Syst.*, vol. 61, no. 4, pp. 398–403, 2013.
[86] S. Goose, S. Güven, X. Zhang, S. Sudarsky, and N. Navab, ''*PARIS*: Fusing vision-based location tracking with standards-based 3D visualization and speech interaction on a PDA,'' in *Proc. IEEE DMS*, Sep. 2004, pp. 75–80.
[87] S. White, S. Feiner, and J. Kopylec, ''Virtual vouchers: Prototyping a mobile augmented reality user interface for botanical species identification,'' in *Proc. IEEE Symp. 3D User Interface*, Mar. 2006, pp. 119–126.
[88] S. Deffeyes, ''Mobile augmented reality in the data center,'' *IBM J. Res. Develop.*, vol. 55, no. 5, pp. 487–491, 2011.
[89] F. K. Wacker *et al.*, ''An augmented reality system for MR image–guided needle biopsy: Initial results in a swine model,'' *Radiology*, vol. 238, no. 2, pp. 497–504, 2006.
[90] M. Rosenthal *et al.*, ''Augmented reality guidance for needle biopsies: An initial randomized, controlled trial in phantoms,'' *Med. Image Anal.*, vol. 6, no. 3, pp. 313–320, 2002.
[91] Glass. (2013). *Google Glass Project*. [Online]. Available: http://www.google.com/glass/start/
[92] Layer. (2013). *LayAR Project*. [Online]. Available: http://www.layar.com/
[93] Wikitude, 2013. *Wikitude: Mobile Augmented Reality Technology Provider for Smartphones, Tablets and Digital Eyewear*. [Online]. Available: www.wikitude.org
[94] J. Grubert, T. Langlotz, and R. Grasset, ''Augmented reality browser survey,'' Inst. Comput. Graph. Vis., Inffeldgasse, Graz, Tech. Rep. 8010, 2011.
[95] A. Morrison *et al.*, ''Collaborative use of mobile augmented reality with paper maps,'' *Comput. Graph.*, vol. 35, no. 4, pp. 798–799, 2011.
[96] J. Irizarry, M. Gheisari, G. Williams, and B. N. Walker, ''InfoSPOT: A mobile augmented reality method for accessing building information through a situation awareness approach,'' *Autom. Construct.*, vol. 33, pp. 11–23, Aug. 2012.
[97] M. J. Lee, Y. Wang, and H. B.-L. Duh, ''AR UX design: Applying AEIOU to handheld augmented reality browser,'' in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR-AMH)*, Nov. 2012, pp. 99–100.
[98] S. Irshad and D. R. A. Rambli, ''Design implications for quality user experience in mobile augmented reality applications,'' in *Advanced Computer and Communication Engineering Technology*. New York, NY, USA: Cham, Switzerland: Springer, 2016, pp. 1283–1294.
[99] S. Irshad and D. R. A. Rambli, ''Multi-layered mobile augmented reality framework for positive user experience,'' in *Proc. 2nd Int. Conf. HCI UX Indonesia*, 2016, pp. 21–26.
[100] S. Mang, ''Towards improving instruction presentation for indoor navigation,'' Dept. Electr. Eng. Inform. Technol., Inst. Med. Technol., München, Germany, Tech. Rep. 80333, 2013.
[101] Z. Huang, P. Hui, and C. Peylo. (2014). ''When augmented reality meets big data.'' [Online]. Available: https://arxiv.org/abs/1407.7223
[102] M. J. Kim, ''A framework for context immersion in mobile augmented reality,'' *Autom. Construct.*, vol. 33, pp. 79–85, Aug. 2013.
[103] W. Hürst and C. Van Wezel, ''Gesture-based interaction via finger tracking for mobile augmented reality,'' *Multimedia Tools Appl.*, vol. 62, no. 1, pp. 233–258, 2013.
[104] H. Seichter, J. Grubert, and T. Langlotz, ''Designing mobile augmented reality,'' in *Proc. 15th Int. Conf. Human-Comput. Interact. Mobile Devices Services*, 2013, pp. 616–621.
[105] M. de Sá and E. F. Churchill, ''Mobile augmented reality: A design perspective,'' in *Human Factors in Augmented Reality Environments*. New York, NY, USA: Springer, 2013, pp. 139–164.
[106] J. D. Bolter, M. Engberg, and B. MacIntyre, ''Media studies, mobile augmented reality, and interaction design,'' *Interactions*, vol. 20, no. 1, pp. 36–45, 2013.
[107] S. Ahn, H. Ko, and B. Yoo, ''Webizing mobile augmented reality content,'' *New Rev. Hypermedia Multimedia—Cyber, Phys. Soc. Comput.*, vol. 20, no. 1, pp. 79–100, 2014. [Online]. Available: http://dx.doi.org/10.1080/13614568.2013.857727
[108] L. Ventä-Olkkonen, M. Posti, O. Koskenranta, and J. Häkkilä, ''Investigating the balance between virtuality and reality in mobile mixed reality UI design: User perception of an augmented city,'' in *Proc. 8th Nordic Conf. Human-Comput. Interact., Fun, Fast, Found.*, 2014, pp. 137–146.
[109] S. Irshad and D. R. A. Rambli, ''User experience evaluation of mobile AR services,'' in *Proc. 12th Int. Conf. Adv. Mobile Comput. Multimedia*, 2014, pp. 119–126.
[110] S. Irshad and D. R. A. Rambli, ''Preliminary user experience framework for designing mobile augmented reality technologies,'' in *Proc. 4th Int. Conf. Interact. Digit. Media (ICIDM)*, 2015, pp. 1–4.
[111] N. M. Khan *et al.*, ''Towards a shared large-area mixed reality system,'' in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6.

[112] P. E. Kourouthanassis, C. Boletsis, and G. Lekakos, "Demystifying the design of mobile augmented reality applications," *Multimedia Tools Appl.*, vol. 74, no. 3, pp. 1045–1066, 2015.

[113] J. T. Doswell and A. Skinner, "Augmenting human cognition with adaptive augmented reality," in *Proc. Int. Conf. Augmented Cognit.*, 2014, pp. 104–113.

[114] G. Reitmayr and D. Schmalstieg, "Location based applications for mobile augmented reality," in *Proc. AUIC*, 2003, pp. 65–73.

[115] D. Wagner and D. Schmalstieg, "ARToolKit on the PocketPC platform," in *Proc. IEEE Int. Augmented Reality Toolkit Workshop*, Oct. 2003, pp. 14–15.

[116] G. Klein and T. Drummond, "Sensor fusion and occlusion refinement for tablet-based AR," in *Proc. ISMAR*, 2004, pp. 38–47.

[117] M. Zöllner, J. Keil, H. Wüst, and D. Pletinckx, "An augmented reality presentation system for remote cultural heritage sites," in *Proc. Virtual Reality, Archaeol. Cultural Heritage*, 2009, pp. 112–126.

[118] M. Mohring, C. Lessig, and O. Bimber, "Video see-through AR on consumer cell-phones," in *Proc. ISMAR*, 2004, pp. 252–253.

[119] S. Vert and R. Vasiu, "Integrating linked data in mobile augmented reality applications," in *Proc. Int. Conf. Inf. Softw. Technol.*, 2014, pp. 324–333.

[120] D. W. F. van Krevelen and R. Poelman, "Augmented reality: Technologies, applications, and limitations," Dept. Comput. Sci., Vrije Univ. Amsterdam, Amsterdam, The Netherlands, Tech. Rep., 2007.

[121] M. Kalkusch, T. Lidy, M. Knapp, G. Reitmayr, H. Kaufmann, and D. Schmalstieg, "Structured visual markers for indoor pathfinding," in *Proc. 1st IEEE Int. Workshop ARToolKit*, Sep. 2002, p. 8.

[122] W. Piekarski, R. Smith, and B. H. Thomas, "Designing backpacks for high fidelity mobile outdoor augmented reality," in *Proc. ISMAR*, 2004, pp. 280–281.

[123] W. Pasman and C. Woodward, "Implementation of an augmented reality system on a PDA," in *Proc. ISMAR*, 2003, pp. 276–277.

[124] T. Pintaric, D. Wagner, F. Ledermann, and D. Schmalstieg, "Towards massively multi-user augmented reality on handheld devices," in *Pervasive Computing*. Berlin, Germany: Springer-Verlag, 2005, pp. 208–219.

[125] G. Reitmayr and T. Drummond, "Going out: Robust model-based tracking for outdoor augmented reality," in *Proc. ISMAR*, 2006, pp. 119–122.

[126] J. Mantyjarvi, F. Paternò, Z. Salvador, and C. Santoro, "Scan and tilt: Towards natural interaction for mobile museum guides," in *Proc. MobileHCI*, 2006, pp. 191–194.

[127] I. Barakonyi and D. Schmalstieg, "Ubiquitous animated agents for augmented reality," in *Proc. ISMAR*, 2006, pp. 145–154.

[128] I. Herbst, A. K. Braun, R. McCall, and W. Broll, "TimeWarp: Interactive time travel with a mobile mixed reality game," in *Proc. MobileHCI*, 2008, pp. 235–244.

[129] J. Gausemeier, J. Fründ, and C. Matysczok, "Development of a process model for efficient content creation for mobile augmented reality applications," in *Proc. CAD*, 2002, pp. 259–270.

[130] J. Gausemeier, J. Fruend, C. Matysczok, B. Bruederlin, and D. Beier, "Development of a real time image based object recognition method for mobile ar-devices," in *Proc. SIGGRAPH AFRIGRAPH*, 2003, pp. 133–139.

[131] P. Renevier, L. Nigay, J. Bouchet, and L. Pasqualetti, "Generic interaction techniques for mobile collaborative mixed systems," in *Proc. CADUI*, 2004, pp. 307–320.

[132] P. Ferdinand, S. Müller, T. Ritschel, and B. Urban, "The eduventure—A new approach of digital game based learning combining virtual and mobile augmented reality game episodes," in *Proc. Workshop Game Based Learn. DeLFI GMW*, vol. 13. 2005, p. 11.

[133] S. Guven, S. Feiner, and O. Oda, "Mobile augmented reality interaction techniques for authoring situated media on-site," in *Proc. ISMAR*, 2006, pp. 235–236.

[134] J. Quarles, S. Lampotang, I. Fischler, P. Fishwick, and B. Lok, "A mixed reality approach for merging abstract and concrete knowledge," in *Proc. VR*, Mar. 2008, pp. 27–34.

[135] S. Han, S. Yang, and J. Kim, "Eyeguardian: A framework of eye tracking and blink detection for mobile device users," in *Proc. HotMobile*, 2012, pp. 27–34.

[136] J. Newman, G. Schall, I. Barakonyi, and A. Schurzinger, "Wide-area tracking tools for augmented reality," in *Proc. Adv. Pervasive Comput.*, 2006, pp. 143–146.

[137] ITACITUS. (2008). *Intelligent Tourism and Cultural Information Through Ubiquitous Services*. [Online]. Available: http://www.itacitus.org

[138] S. Kang *et al.*, "SeeMon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments," in *Proc. MobiSys*, 2008, pp. 267–280.

[139] E. Kruijff, E. Mendez, E. Veas, and T. Gruenewald, "On-site monitoring of environmental processes using mobile augmented reality (hydrosys)," in *Proc. Infrastruct. Platforms Workshop*, 2010, pp. 1–7.

[140] A. Henrysson and M. Ollila, "Umar: Ubiquitous mobile augmented reality," in *Proc. 3rd Int. Conf. Mobile Ubiquitous Multimedia*, 2004, pp. 41–45.

[141] K. Greene, "Hyperlinking reality via phones," *MIT Technol. Rev.*, Nov./Dec. 2006. [Online]. Available: https://www.technologyreview.com/s/406899/hyperlinking-reality-via-phones/

[142] M. Rohs, "Marker-based embodied interaction for handheld augmented reality games," *J. Virtual Reality Broadcast.*, vol. 4, no. 5, pp. 618–639, 2007.

[143] D. Schmalstieg and D. Wagner, "Mobile phones as a platform for augmented reality," in *Proc. IEEE VR Workshop Softw. Eng. Archit. Realtime Interact. Syst.*, 2008, pp. 43–44.

[144] G. Takacs, V. Chandrasekhar, and N. Gelfand, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proc. MIR*, 2008, pp. 427–434.

[145] D. M. Chen, S. S. Tsai, R. Vedantham, R. Grzeszczuk, and B. Girod, "Streaming mobile augmented reality on mobile phones," in *Proc. ISMAR*, Oct. 2009, pp. 181–182.

[146] R. Paucher and M. Turk, "Location-based augmented reality on mobile phones," in *Proc. CVPR*, Jun. 2010, pp. 9–16.

[147] T. Langlotz *et al.*, "Sketching up the world: *In situ* authoring for mobile augmented reality," in *Pers. Ubiquitous Comput.*, vol. 16, no. 6, pp. 623–630, Aug. 2012.

[148] G. Reitmayr and D. Schmalstieg, "Mobile collaborative augmented reality," in *Proc. ISAR*, Oct. 2001, pp. 114–123.

[149] G. Reitmayr and D. Schmalstieg, "An open software architecture for virtual reality interaction," in *Proc. ACM Symp. Virtual Reality Softw. Technol.*, 2001, pp. 47–54.

[150] G. Reitmayr and D. Schmalstieg, "OpenTracker—An open software architecture for reconfigurable tracking based on XML," in *Proc. IEEE Virtual Reality*, Mar. 2001, pp. 285–286.

[151] D. Wagner and D. Schmalstieg, "Muddleware for prototyping mixed reality multiuser games," in *Proc. IEEE Virtual Reality*, Mar. 2007, pp. 235–238.

[152] F. Hohl, U. Kubach, A. Leonhardi, K. Rothermel, and M. Schwehm, "Nexus—An open global infrastructure for spatial aware applications," in *Proc. Mobile Comput. Netw.*, 1999, pp. 249–255.

[153] D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang, "A model-based, open architecture for mobile, spatially aware applications," in *Proc. 7th Int. Symp. Adv. Spatial Temporal Databases*, 2001, pp. 392–401.

[154] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. IWAR*, Oct. 1999, pp. 85–94.

[155] W. Piekarski and B. H. Thomas, "Tinmith-evo5: A software architecture for supporting research into outdoor augmented reality environments," Wearable Comput. Lab., University of South Australia, Mawson Lakes, SA, Australia, Tech. Rep., 2001.

[156] W. Piekarski and H. T. Bruce, "An object-oriented software architecture for 3D mixed reality applications," in *Proc. ISMAR*, 2003, pp. 247–256.

[157] W. Piekarski and B. H. Thomas, "Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer," in *Proc. Wearable Comput.*, Oct. 2001, pp. 31–38.

[158] M. Bauer *et al.*, "Design of a component-based augmented reality framework," in *Proc. Augmented Reality*, 2001, pp. 45–54.

[159] M. Tonnis, "Data management for augmented reality applications," M.S. thesis, Tech. Univ. Munich, Munich, Germany, 2003.

[160] B. Bruegge and G. Klinker. (2005). "Dwarf-distributed wearable augmented reality framework." [Online]. Available: http://ar.in.tum.de/Chair/DwarfWhitePaper

[161] A. Hill, B. MacIntyre, M. Gandy, B. Davidson, and H. Rouzati, "KHARMA: An open KML/HTML architecture for mobile augmented reality applications," in *Proc. ISMAR*, Oct. 2010, pp. 233–234.

[162] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *Proc. CVWW*, 2007, pp. 233–234.

[163] A. Kirillov. (2010). "Glyph's recognition." [Online]. Available: http://www.aforgenet.com/articles/glyph_recognition/

[164] P. Savioja, P. Järvinen, T. Karhela, P. Siltanen, and C. Woodward, "Developing a mobile, service-based augmented reality tool for modern maintenance work," in *Proc. Virtual Reality HCII*, 2007, pp. 285–288.

[165] P. Siltanen, T. Karhela, C. Woodward, and P. Savioja, "Augmented reality for plant lifecycle management," in *Proc. Concurrent Enterprising*, 2007, pp. 285–288.

[166] P. Välkkynen, A. Boyer, T. Urhemaa, and R. Nieminen, "Mobile augmented reality for retail environments," in *Proc. Mobile HCI Workshop Mobile Interact. Retail Environ.*, 2011, pp. 1–4.

[167] Z. Huang, W. Li, P. Hui, and C. Peylo, "CloudRidAR: A cloud-based architecture for mobile augmented reality," in *Proc. Workshop Mobile Augmented Reality Robot. Technol.-Based Syst.*, 2014, pp. 29–34.

[168] A. Mossel and C. Schönauer, G. Gerstweiler, and H. Kaufmann, "ARTiFICe: Augmented reality framework for distributed collaboration," *Int. J. Virtual Reality*, vol. 11, no. 3, pp. 1–7, 2013.

[169] (2010). "AndAR." [Online]. Available: https://code.google.com/p/andar/

[170] (2011). "DroidAR." [Online]. Available: https://code.google.com/p/droidar/

[171] (2013). "GRATF[Online]. Available: http://www.aforgenet.com/projects/gratf/

[172] S. Feiner, B. MacIntyre, and T. Höllerer, "Wearing it out: First steps toward mobile augmented reality systems," in *Proc. Mixed Reality, Merging Real Virtual Worlds*, 1999, pp. 363–377.

[173] G. Reitmayr and D. Schmalstieg, "Data management strategies for mobile augmented reality," in *Proc. Int. Workshop Softw. Technol. Augmented Reality Syst.*, 2003, pp. 47–52.

[174] J. Schöning, M. Rohs, and S. Kratz, "Map torchlight: A mobile augmented reality camera projector unit," in *Proc. CHI*, 2009, pp. 3841–3846.

[175] J. Zhou, I. Lee, B. H. Thomas, A. Sansome, and R. Menassa, "Facilitating collaboration with laser projector-based spatial augmented reality in industrial applications," in *Recent Trends of Mobile Collaborative Augmented Reality Systems*, 1st ed. London, U.K.: Springer, 2011, pp. 161–173.

[176] P. Mistry, P. Maes, and L. Chang, "WUW-wear Ur world: A wearable gestural interface," in *Proc. Human Factors Comput. Syst.*, 2009, pp. 4111–4116.

[177] Project3D. (2011). *How to Project on 3D Geometry*. [Online]. Available: http://vvvv.org/documentation/how-to-project-on-3d-geometry

[178] J. P. Rolland, R. L. Holloway, H. Fuchs, "Comparison of optical and video see-through, head-mounted displays," *Proc. SPIE*, vol. 2351, pp. 293–307, Dec. 2004. [Online]. Available: http://dx.doi.org/10.1117/12.197322

[179] P. Lang, A. Kusej, A. Pinz, and G. Brasseur, "Inertial tracking for mobile augmented reality," in *Proc. Instrum. Meas.*, May 2002, pp. 1583–1587.

[180] C. Randell, C. Djiallis, and H. Müller, "Personal position measurement using dead reckoning," in *Proc. Wearable Comput.*, vol. 2003, pp. 166–173.

[181] D. Hallaway, S. Feiner, and T. Höllerer, "Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality," *Appl. Artif. Intell., Int. J.*, vol. 18, no. 6, pp. 477–500, Jul. 2004.

[182] S. Diverdi and T. Hollerer, "GroundCam: A tracking modality for mobile mixed reality," in *Proc. VR*, Mar. 2007, pp. 75–82.

[183] J. P. Rolland, Y. Baillot, and A. Goon, "A survey of tracking technology for virtual environments," in *Proc. Wearable Comput. Augumented Reality*, 2001, pp. 67–112.

[184] J. Wither and T. Hollerer, "Depth cues for outdoor augmented reality," in *Proc. Wearable Comput.*, Oct. 2005, pp. 92–99.

[185] T. Höllerer, J. Wither, and S. DiVerdi, "'Anywhere augmentation': Towards mobile augmented reality in unprepared environments," in *Location Based Services and TeleCartography*, G. Gartner, W. Cartwright, and M. P. Peterson, Eds. Berlin, Germany: Springer, 2007, pp. 393–416, doi: 10.1007/978-3-540-36728-4_29, [Online]. Available: http://dx.doi.org/10.1007/978-3-540-36728-4_29

[186] P. Belimpasakis, P. Selonen, and Y. You, "Bringing user-generated content from internet services to mobile augmented reality clients," in *Proc. Cloud-Mobile Converg. Virtual Reality*, Mar. 2010, pp. 14–17.

[187] J. Chung, M. Donahoe, and C. Schmandt, "Indoor location sensing using geo-magnetism," in *Proc. MobiSys*, 2011, pp. 141–154.

[188] Z. Sun, A. Purohit, S. Pan, F. Mokaya, R. Bose, and P, Zhang, "Polaris: Getting accurate indoor orientations for mobile devices using ubiquitous visual patterns on ceilings," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, p. 14.

[189] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Indoor wayfinding: Developing a functional interface for individuals with cognitive impairments," in *Proc. 8th Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2006, pp. 95–102.

[190] W. C. Chung and D. Ha, "An accurate ultra wideband (UWB) ranging for precision asset location," in *Proc. IEEE Conf. Ultra Wideband Syst. Technol.*, Nov. 2003, pp. 389–393.

[191] S. Gezici *et al.*, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, Jul. 2005.

[192] J. Gonzalez *et al.*, "Combination of UWB and GPS for indoor-outdoor vehicle localization," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Oct. 2007, pp. 1–7.

[193] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2003, pp. 407–415.

[194] S. Goose, H. Wanning, and G. Schneider, "*Mobile reality*: A PDA-based multimodal framework synchronizing a hybrid tracking solution with 3D graphics and location-sensitive speech interaction," in *Proc. Int. Conf. Ubiquitous Comput.*, 2002, pp. 33–47.

[195] D. Hallaway, T. Hollerer, and S. Feiner, "Coarse, inexpensive, infrared tracking for wearable computing," in *Proc. 7th IEEE Int. Symp. Wearable Comput.*, Oct. 2003, pp. 69–78.

[196] D. C. McFarlane and S. M. Wilder, "Interactive dirt: Increasing mobile work performance with a wearable projector-camera system," in *Proc. UbiComp*, 2009, pp. 205–214.

[197] Y. Ishiguro, A. Mujibiya, T. Miyaki, and J. Rekimoto, "Aided eyes: Eye activity sensing for daily life," in *Proc. 1st Augmented Human Int. Conf.*, 2010, pp. 1–7.

[198] L. Alto, N. Göthlin, J. Korhonen, and T. Ojala, "Bluetooth and WAP push based location-aware mobile advertising system," in *Proc. MobiSys*, 2004, pp. 49–58.

[199] M. Rauhala, A.-S. Gunnarsson, and A. Henrysson, "A novel interface to sensor networks using handheld augmented reality," in *Proc. MobileHCI*, 2006, pp. 145–148.

[200] M. S. Bargh and R. de Groote, "Indoor localization based on response rate of bluetooth inquiries," in *Proc. Mobile Entity Localization Tracking GPS-Less Environm.*, 2008, pp. 49–54.

[201] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proc. MobiSys*, 2010, pp. 299–314.

[202] R. Wang, F. Zhao, H. Luo, B. Lu, and T. Lu, "Fusion of Wi-Fi and bluetooth for indoor localization," in *Proc. LMBS*, 2011, pp. 63–66.

[203] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 32–43.

[204] E. Foxlin and M. Harrington, "WearTrack: A self-referenced head and hand tracker for wearable computers and portable VR," in *Proc. ISWC*, Oct. 2000, pp. 155–162.

[205] J. Newman, D. Ingram, and A. Hopper, "Augmented reality in a wide area sentient environment," in *Proc. ISMAR*, Oct. 2001, pp. 77–86.

[206] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 590–596.

[207] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst, "Shape recognition and pose estimation for mobile augmented reality," in *Proc. ISMAR*, Oct. 2009, pp. 65–71.

[208] U. Neumann and S. You, "Natural feature tracking for augmented reality," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 53–64, Mar. 1999.

[209] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[210] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, vol. 3951. 2006, pp. 404–417. [Online]. Available: http://dx.doi.org/10.1007/11744023_32

[211] I. Skrypnyk and D. G. Lowe, "Scene modelling, recognition and tracking with invariant image features," in *Proc. ISMAR*, Nov. 2004, pp. 110–119.

[212] G. Fritz, C. Seifert, and L. Paletta, "A mobile vision system for urban detection with informative local descriptors," in *Proc. ICVS*, Jan. 2006, p. 30.

[213] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[214] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2004, pp. II-506–II-513.
[215] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.
[216] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proc. ISMAR*, Sep. 2008, pp. 125–134.
[217] G. Takacs, V. Chandrasekhar, B. Girod, and R. Grzeszczuk, "Feature tracking for mobile augmented reality using video coder motion vectors," in *Proc. ISMAR*, Nov. 2007, pp. 141–144.
[218] W.-C. Chen, Y. Xiong, J. Gao, N. Gelfand, and R. Grzeszczuk, "Efficient extraction of robust image features on mobile devices," in *Proc. ISMAR*, 2007, pp. 287–288.
[219] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. CVPR*, 2009, pp. 2937–2944.
[220] D. Wagner, D. Schmalstieg, and H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," in *Proc. ISMAR*, 2009, pp. 57–64.
[221] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 3, pp. 355–368, May/Jun. 2010.
[222] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.
[223] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2548–2555.
[224] M. Li, S. Rane, and P. Boufounos, "Quantized embeddings of scale-invariant image features for mobile augmented reality," in *Proc. MMSP*, 2012, pp. 1–6.
[225] J. S. Park, B. J. Bae, and R. Jain. (2012). *Fast Natural Feature Tracking for Mobile Augmented Reality Applications*. [Online]. Available: http://onlinepresent.org/proceedings/vol3_2012/84.pdf
[226] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2538–2551, Oct. 2014.
[227] Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 678–691.
[228] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2427–2434.
[229] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 723–730.
[230] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1177–1184.
[231] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 49–56.
[232] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng, "Blurred target tracking by blur-driven tracker," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1100–1107.
[233] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.
[234] J. Li *et al.*, "A hybrid pose tracking approach for handheld augmented reality," in *Proc. 9th Int. Conf. Distrib. Smart Cameras*, 2015, pp. 7–12.
[235] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, vol. 32. 2014, pp. I-647–I-655.
[236] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.
[237] *Sh1r0. Caffe-Android-Lib*, accessed on Mar. 1, 2017. [Online]. Available: https://github.com/sh1r0/caffe-android-lib
[238] M. Hashemi. *CNNdroid*, accessed on Mar. 1, 2017. [Online]. Available: https://github.com/ENCP/CNNdroid
[239] R. Behringer, J. Park, and V. Sundareswaran, "Model-based visual tracking for outdoor augmented reality applications," in *Proc. ISMAR*, 2002, pp. 277–278.
[240] B. Jiang, U. Neumann, and S. You, "A robust hybrid tracking system for outdoor augmented reality," in *Proc. VR*, 2004, pp. 3–10.
[241] Z. Hu and K. Uchimura, "Fusion of vision, GPS and 3D gyro data in solving camera registration problem for direct visual navigation," *Int. J. ITS Res.*, vol. 4, no. 1, pp. 3–12, 2006.
[242] P. Honkamaa, S. Siltanen, J. Jäppinen, C. Woodward, and O. Korkalo, "Interactive outdoor mobile augmentation using markerless tracking and GPS," in *Proc. VRIC*, 2007, pp. 285–288.
[243] J. Dolezal, Z. Becvar, and T. Zeman, "Performance evaluation of computation offloading from mobile device to the edge of mobile network," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2016, pp. 1–7.
[244] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl. (MobiSys)*, New York, NY, USA, 2014, pp. 68–81. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594383
[245] T. Blajić, D. Nogulić, and M. Družijanić, "Latency improvements in 3G long term evolution," in *Proc. Int. Conv. Inf. Commun. Technol., Electron. Microelectron.*, 2006, pp. 1–6.
[246] M. Abrash. *Latency—The Sine QUA non of AR and VR*, accessed on Feb. 23, 2017. [Online]. Available: http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/
[247] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, USA, 2015, pp. 155–168. [Online]. Available: http://doi.acm.org/10.1145/2809695.2809711
[248] Y. Wu, M. E. Choubassi, and I. Kozintsev, "Augmenting 3D urban environment using mobile devices," in *Proc. ISMAR*, Oct. 2011, pp. 241–242.
[249] D. J. McCaffery and J. Finney, "The need for real time consistency management in P2P mobile gaming environments," in *Proc. ACM SIGCHI Int. Conf. Adv. Comput. Entertainment Technol.*, 2004, pp. 203–211.
[250] N. Joseph, F. Friedrich, and G. Schall, "Construction and maintenance of augmented reality environments using a mixture of autonomous and manual surveying techniques," in *Proc. Opt. 3-D Meas. Techn.*, 2005, pp. 1–9.
[251] T. Hossmann, C. Efstratiou, and C. Mascolo, "Collecting big datasets of human activity one checkin at a time," in *Proc. HotPlanet*, 2012, pp. 15–20.
[252] (2005). "Khronos." [Online]. Available: http://www.khronos.org/opengles
[253] (2008). "Khronos." [Online]. Available: http://www.khronos.org/opencl/
[254] C. Wutte, "Computer vision on mobile phone GPUs," M.S. thesis, Tech. Univ. Graz, Tech. Univ., Berlin, Berlin, Germany, 2009.
[255] G. R. Kayombya, "SIFT feature extraction on a Smartphone GPU using OpenGL ES2.0," M.S. thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2010.
[256] N. Singhal, I. K. Park, and S. Cho, "Implementation and optimization of image processing algorithms on handheld GPU," in *Proc. ICIP*, Sep. 2010, pp. 4481–4484.
[257] N. Singhal, J. W. Yoo, and H. Y. Choi, "Design and optimization of image processing algorithms on mobile GPU," in *Proc. SIGGRAPH*, 2011, p. 21.
[258] R. Hofmann, "Extraction of natural feature descriptors on mobile GPUs," M.S. thesis, Dept. Comput. Sci., Univ. Koblenz Landau, Mainz, Germany, 2012.
[259] J. Leskela, J. Nikula, and M. Salmela, "Opencl embedded profile prototype in mobile device," in *Proc. SiPS*, 2009, pp. 279–284.
[260] A.-R. Baek, K. Lee, and H. Choi, "CPU and GPU parallel processing for mobile augmented reality," in *Proc. 6th Int. Congr. Image Signal Process. (CISP)*, vol. 1. Dec. 2013, pp. 133–137.
[261] D. Wagner and D. Schmalstieg, "Making augmented reality practical on mobile phones," *IEEE Comput. Graph. Appl.*, vol. 29, no. 3, pp. 12–15, Jul./Aug. 2009.
[262] Imagination Technologies Limited, "PowerVR series 5 architecture guide for developers," Imagination, Kings Langley, U.K., Tech. Rep., Mar. 2016. [Online]. Available: http://cdn.imgtec.com/sdk-documentation/PowerVR+Series5.Architecture+Guide+for+Developers.pdf

[263] J. M. Arnau, J. M. Parcerisa, and P. Xekalakis, "Boosting mobile GPU performance with a decoupled access/execute fragment processor," in *Proc. ISCA*, Jun. 2012, pp. 84–93.

[264] K. T. Cheng and Y.-C. Wang, "Using mobile GPU for general-purpose computing—A case study of face recognition on smartphones," in *Proc. VLSI-DAT*, Apr. 2011, pp. 1–4.

[265] J. Owens, "Streaming architectures and technology trends," in *ACM SIGGRAPH Courses*. New York, NY, USA: ACM, 2005, pp. 457–470. [Online]. Available: http://doi.acm.org/10.1145/1198555.1198766

[266] T. Akenine-Möller and J. Ström, "Graphics processing units for handhelds," in *Proc. IEEE*, vol. 96, no. 5, pp. 779–789, May 2008.

[267] T. Capin, K. Pulli, and T. Akenine-Möller, "The state of the art in mobile graphics research," *IEEE Comput. Graph. Appl.*, vol. 28, no. 4, pp. 74–84, Jul. 2008.

[268] T. Akenine-Möller and J. Ström, "Graphics for the masses: A hardware rasterization architecture for mobile phones," in *Proc. SIGGRAPH*, 2003, pp. 801–808.

[269] J. Ström and T. Akenine-Möller, "ipackman: High-quality, low-complexity texture compression for mobile phones," in *Proc. Graph. Hardw.*, 2005, pp. 63–70.

[270] F. Duguet and G. Drettakis, "Flexible point-based rendering on mobile devices," *IEEE Comput. Graph. Appl.*, vol. 24, no. 4, pp. 57–63, Jul. 2004.

[271] V. Setlur, X. J. Chen, and Y. Q. Xu, "Retargeting vector animation for small displays," in *Proc. Mobile Ubiquitous Multimedia*, 2005, pp. 69–77.

[272] J. Hasselgren and T. Akenine-Möller, "PCU: The programmable culling unit," *ACM Trans. Graph.*, vol. 26, no. 3, p. 92, 2007.

[273] H. Fuchs *et al.*, "Pixel-planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories," in *Proc. SIGGRAPH*, 1989, pp. 79–88.

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

[275] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. SIGGRAPH*, 1995, pp. 39–46.

[276] Y. X. Gu, N. Li, and L. Chang, "Recent trends of mobile collaborative augmented reality systems," in *A Collaborative Augmented Reality Networked Platform for Edutainment*, 1st ed. London, U.K.: Springer, 2011, pp. 117–125.

[277] F. Lamberti, C. Zunino, and A. Sanna, "An accelerated remote graphics architecture for PDAS," in *Proc. SIGGRAPH Web 3D Symp.*, 2003, pp. 51–61.

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

[279] X. Luo, "From augmented reality to augmented computing: A look at cloud-mobile convergence," in *Proc. ISUVR*, 2009, pp. 29–32.

[280] Y. Lu, S. Li, and H. Shen, "Virtualized screen: A third element for cloud-mobile convergence," *IEEE Multimedia Mag.*, vol. 18, no. 2, pp. 4–11, Feb. 2011.

[281] C. Wang and Z. Li, "Parametric analysis for adaptive computation offloading," *ACM SIGPLAN Notices*, vol. 39, no. 6, pp. 119–130, 2004.

[282] R. Wolski, S. Gurun, and C. Krintz, "Using bandwidth data to make computation offloading decisions," in *Proc. PDPS*, 2008, pp. 1–8.

[283] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate computing: A survey," *IEEE Des. Test*, vol. 33, no. 1, pp. 8–22, Feb. 2016.

[284] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, p. 62, 2016.

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

[286] T. Moreau *et al.*, "SNNAP: Approximate computing on programmable SoCs via neural acceleration," in *Proc. IEEE 21st Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2015, pp. 603–614.

[287] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proc. 50th Annu. Design Autom. Conf.*, 2013, p. 113.

[288] V. Vassiliadis *et al.*, "Exploiting significance of computations for energy-constrained approximate computing," *Int. J. Parallel Programm.*, vol. 44, no. 5, pp. 1078–1098, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10766-016-0409-6

[289] A. Sampson *et al.*, "ACCEPT: A programmer-guided compiler framework for practical approximate computing," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep. UW-CSE-15-01-01, 2015.

[290] J. Paczkowski. (2009). *iPhone Owners Would Like to Replace Battery*. [Online]. Available: http://digitaldaily.allthingsd.com/20090821/iphone-owners-would-like-to-replace-battery-att

[291] N. V. Rodriguez and J. Crowcroft, "Energy management techniques in modern mobile handsets," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 179–198, 1st Quart., 2013.

[292] "The benefits of multiple CPU cores in mobile devices," NVIDIA, Santa Clara, CA, USA, Tech. Rep., 2010. [Online]. Available: https://www.nvidia.com/content/PDF/tegra_white_papers/Benefits-of-Multi-core-CPUs-in-Mobile-Devices_Ver1.2.pdf

[293] J.-H. Sohn, Y.-H. Park, C.-W. Yoon, R. Woo, S.-J. Park, and H.-J. Yoo, "Low-power 3D graphics processors for mobile terminals," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 90–99, Dec. 2005.

[294] B. C. Mochocki, K. Lahiri, and S. Cadambi, "Signature-based workload estimation for mobile 3D graphics," in *Proc. Design Autom.*, 2006, pp. 592–597.

[295] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[296] S. Kosta, A. Aucinas, and P. Hui, "ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading," in *Proc. INFOCOM*, 2012, pp. 945–953.

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

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

[299] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen, "Comet: Code offload by migrating execution transparently," in *Proc. 10th USENIX (OSDI)*, 2012, pp. 93–106.

[300] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[301] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Proc. IEEE INFOCOM*, May 2014, pp. 1060–1068.

[302] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," in *Proc. IEEE 16th Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Sep. 2012, pp. 123–132.

[303] P. A. Rego, P. B. Costa, E. F. Coutinho, L. S. Rocha, F. A. Trinta, and J. N. de Souza, "Performing computation offloading on multiple platforms," *Comput. Commun.*, vol. 105, pp. 1–13, Jun. 2016.

[304] E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using mapreduce," School Comput. Sci., Carnegie Mellon Univ., Tech. Rep. CMU-CS-09-164, 2009.

[305] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *Proc. MobiHoc*, 2012, pp. 145–154.

[306] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, "OPENRP: A reputation middleware for opportunistic crowd computing," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 115–121, Jul. 2016.

[307] J. Herling and W. Broll, "An adaptive training-free feature tracker for mobile phones," in *Proc. VRST*, 2010, pp. 35–42.

[308] A. LaMarca, Y. Chawathe, and S. Consolvo, "Place lab: Device positioning using radio beacons in the wild pervasive computing," in *Proc. Int. Conf. Pervasive Comput.*, 2005, pp. 301–306.

[309] R. El Hattachi and J. Erfanian, "5G white paper," Northern Alliance, Next Generat. Mobile Netw., Frankfurt, Germany, White Paper, 2015.

[310] R. Ford, M. Zhang, M. Mezzavilla, S. Dutta, S. Rangan, and M. Zorzi. (Feb. 2016). "Achieving ultra-low latency in 5G millimeter wave cellular networks." [Online]. Available: https://arxiv.org/abs/1602.06925

[311] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

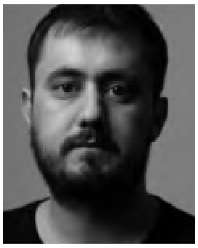
[312] A. Acquisti, R. Gross, and F. Stutzman, "Faces of Facebook: Privacy in the age of augmented reality," Tech. Rep., 2012.

[313] J. Shu, R. Zheng, and P. Hui. (Oct. 2016). "Cardea: Context-aware visual privacy protection from pervasive cameras." [Online]. Available: https://arxiv.org/abs/1610.00889

[314] J. Shu, R. Zheng, and P. Hui, "Demo: Interactive visual privacy control with gestures," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Services Companion (MobiSys)*, New York, NY, USA, 2016, p. 120. [Online]. Available: http://doi.acm.org/10.1145/2938559.2938571

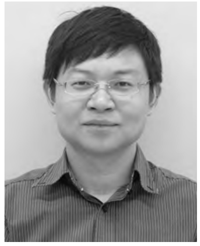
[315] [Online]. Available: http://news.softpedia.com/news/In-Stat-Research-Predicts-Multi-Core-CPUs-in-88-Of-Mobile-Platforms-By-2013-129283.shtml

[316] G. Takacs, M. El Choubassi, Y. Wu, and I. Kozintsev, "3D mobile augmented reality in urban scenes," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2011, pp. 1–4.



DIMITRIS CHATZOPOULOS received the Diploma and M.sc. degrees in computer engineering and communications from the Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology and a member of the Symlab. His main research interests are in the areas of mobile augmented reality, mobile computing, device–to–device ecosystems and cryptocurrencies.

CARLOS BERMEJO received the M.Sc. degree in telecommunication engineering from Oviedo university, Spain, in 2012. He is currently pursuing the Ph.D. degreewith The Hong Kong University of Science and Technology. He is with the Symlab Research Group. His main research interests are Internet-of-Things, mobile augmented reality, network security, human-computer-interaction, social networks, and device–to–device communication.



ZHANPENG HUANG received the bachelor's and Ph.D. degrees from Beihang University. After his graduation, he became a Post-Doctor with HKUST-DT System and Media Laboratory (SyMLab), The Hong Kong University of Science and Technology. His research interests include mobile augmented reality, computer-human interaction, virtual reality, and physically-based simulation.

PAN HUI received the Ph.D. degree from the Computer Laboratory, University of Cambridge, and the M.Phil. and B.Eng. degrees from the Department of Electrical and Electronic Engineering, The University of Hong Kong. He is currently a Faculty Member with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, where he directs the HKUST-DT System and Media Lab. He also serves as a Distinguished Scientist with the Telekom Innovation Laboratories (T-labs), Germany and an Adjunct Professor of social computing and networking with Aalto University, Finland. Before returning to Hong Kong, he spent several years in T-labs and Intel Research Cambridge. He has authored over 150 research papers and has some granted and pending European patents. He has Founded and Chaired several IEEE/ACM conferences/workshops, and has been serving on the organizing and technical program committee of numerous international conferences and workshops, including the ACM SIGCOMM, the IEEE Infocom, the ICNP, the SECON, the MASS, the Globe- com, WCNC, the ITC, the ICWSM, and WWW. He is an Associate Editor for the IEEE Transactions on Mobile Computing and the IEEE Transactions on Cloud Computing, and an ACM Distinguished Scientist.