Browsing the Web of Things in Mobile Augmented Reality

Thomas Zachariah
University of California, Berkeley
Berkeley, California
tzachari@berkeley.edu

Prabal Dutta
University of California, Berkeley
Berkeley, California
prabal@berkeley.edu

ABSTRACT

With the current augmented reality and low-power radio technology present on mobile platforms, we can imagine a standard and physically tangible browsing mechanism for objects in the Web of Things. We explore a model for user interaction with IoT devices that makes use of mobile augmented reality to allow users to identify new devices or easily access regularly-used devices in their environment, enables immediate interaction with quickly-obtainable user interfaces from the web, and provides developers a convenient platform to display custom interfaces for their devices. This model represents a step towards software-based interaction that might, one day, feel as intuitive, accessible, and familiar as the physical interfaces we commonly encounter in our daily lives.

CCS CONCEPTS

• **Human-centered computing → Web-based interaction**; **User interface design**; **Mixed / augmented reality**; *Ubiquitous and mobile computing systems and tools*;
• **Information systems → Service discovery and interfaces**; *Web interfaces*; Users and interactive retrieval;
• **Networks** → Mobile ad hoc networks.

KEYWORDS

Internet of Things, Mobile, Augmented Reality, Device Discovery, Web Browsing, User Interfaces

ACM Reference Format:
Thomas Zachariah and Prabal Dutta. 2019. Browsing the Web of Things in Mobile Augmented Reality. In *The 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19), February 27–28, 2019, Santa Cruz, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3301293.3302359

HotMobile '19, February 27–28, 2019, Santa Cruz, CA, USA

ACM ISBN 978-1-4503-6273-3/19/02.
https://doi.org/10.1145/3301293.3302359

1 INTRODUCTION

As the Internet of Things (IoT) grows, the issue of accessibility becomes more apparent. The benefits of connecting everyday things to the Internet are limited by a prevailing walled-garden approach to accessing the services provided by each device. Currently, most consumer IoT devices require installation of a separate app on a mobile phone or tablet to enable any form of user interaction. This is reminiscent of the heyday of America Online, whose business model of providing separately-packaged, single-purpose online services thrived for a brief period, but ultimately limited the potential content and resources that a browser with open access to the World Wide Web would later provide. This approach has been moderately successful in piquing initial interest and enabling an entry-point for societal acceptance of the new technology, but it limits the ability for discovery, prevents regular engagement from users, and discourages developers from creating useful services and products.

This issue, of course, has not gone entirely unnoticed. Major players have come together to create a variety of standards, most of which struggle to become widely adopted by developers. Apple and Google have introduced standards in which devices are organized into a "home" and can be managed with a unified controller on a mobile phone, tablet, or smart speaker. This single scoping limits which devices are accessible, what controls are provided, and how data is displayed in the user interface. Additionally, it requires users to make their own associations with devices and utilize recall more often than recognition when trying to interact with a specific device. It also limits the discovery of new "things" in their environment.

With the rising prevalence of reliable and capable mobile augmented reality (MAR) on personal phones and tablets, we may already have the tools necessary to develop a solution that provides a more intuitive approach to discovery, a more tangible approach to interaction, and a more creative approach to presentation. This may also lay the groundwork for a compelling, practical, and useful application for MAR.

In this paper, we examine a potential model for user interaction with IoT devices that makes use of MAR to allow users to identify new devices and easily access regularly-used devices in the physical space of their environment. It could enable immediate interaction with quickly-obtainable user interfaces (UIs) from the web, and provide developers with a convenient platform to display, within perceived physical space, custom interfaces for their devices that can be created using standard web tools. This mobile augmented reality browsing concept is depicted in Figure 1. In our study, we consider a few of the driving applications that could be supported and we explore some of the opportunities and challenges that arise in the development of such a system.

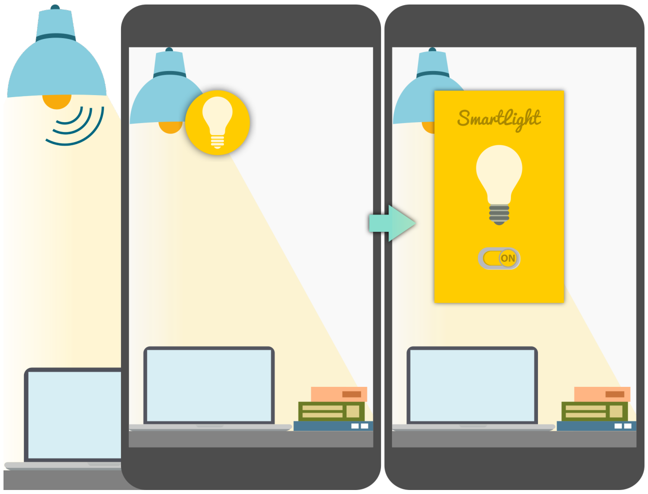

Figure 1: Concept overview. In this model, users can open a "browser" on their smartphone or tablet, which uses the camera to identify devices and discover their associated web interfaces in physical space. When an interface is opened, it can use a JavaScript Bluetooth API or network protocol to interact with the device.

2 RELATED WORK

Interaction Standards for IoT Devices. Device-specific apps are the current standard for interaction in the Internet of Things. They allow fine-grain setup and control of specific devices in an interface aesthetic of the developer's choosing. But this requires knowledge of the device and download of its app. It is also typically only accessible by the owners of the device, which discourages opportunities for ephemeral discovery and interaction.

Unified Control on Mobile Platforms. Unified control systems like Apple Home and Google Home begin to alleviate the problem by providing a single interface to control multiple devices in a house [1, 7]. As their names indicate, they are primarily intended for use in the home and limit scope to devices that implement specific proprietary protocols and are owned by the user. This still often requires download of the devices' apps for setup. It also imposes a standard aesthetic and provides a limited offering of control interfaces (e.g switches and sliders).

Device Discovery. Google's Physical Web allows discovery of nearby devices via Bluetooth Low Energy [6]. The device broadcasts a URL which a phone can detect and open in a browser. The URL can point to a web page that acts as an interface for the device. If the device is connected to the Internet or the page uses a JavaScript Bluetooth API, users can interact with the device in real time. This enables immediate interaction in an interface of the developer's choosing. Unfortunately as the number of devices scales up, the listing model becomes overwhelming and it is often difficult for users to map the appropriate interface to device. This may be, in part, why Google has removed Physical Web support on Android and iOS. However, the concept of associating nearby physical devices with web content via a broadcast service may be useful in the context of an MAR browser that more tangibly connects interfaces to devices.

Early Inspirations for AR-Like Modalities. In 1987, a Bell Labs study on human-system communication discovered that users struggle to identify canonical names and keyword commands for systems using voice or type interfaces [5], indicating that visual articulation might be more effective. This problem continues today in IoT, where users must identify available devices, brands, and associated apps prior to any form of interaction, even with voice-control agents. A 2000 Microsoft study on emerging interfaces for smart devices examined user experiences of touch and speech interfaces [3]. While the study did not discuss AR as a modality, it suggested that the functionality of systems could improve when interfaces are reinforced with an input that provides location-awareness (e.g. gaze or gesture) to help disambiguate a target device.

AR and MAR in Consumer Space. Recently, Apple and Google released AR software development kits for mobile platforms, ARKit and ARCore [1, 7]. With native support for AR and more capable hardware on smartphones and tablets, mobile augmented reality has become a practical reality and exposes the technology to a large user-base. Prior to this, third-party MAR frameworks existed, but much work in AR tended to focus on head-worn display devices rather than mobile. Still, surveys of existing consumer AR technologies from the last decade indicate that while industry interest in AR is high, it lacks compelling practical applications [15, 22].

MAR Systems in Practice. MAR systems designed to enable interaction with objects in an environment usually only work with a specific set of devices, and limit the type of controls devices can have and how they are presented to the user. Some of the most popular MAR applications work only in highly specific environments like an outfitted office, classroom, or exhibit [2, 9, 17]. Many of these employ computer vision with limited training sets, which is made easier with built-in machine learning frameworks on mobile platforms, like Google's ML Kit and Apple's Core ML [1, 7]. HP Reveal is an example that uses image recognition to identify objects "anywhere", but requires the user to manually narrow the search-space by subscribing to a "channel" of objects [18]. It is primarily used to demonstrate object identification, rather than interaction.

Target Identifiers Standard visual identifiers like QR codes or AprilTags are often used for general use cases. The MAR browser could potentially make use of such identifiers, but they may be obtrusive or distracting in the long run. They still generally require users to be somewhat informed about the environment around them and limit discovery when markers are not visible on screen. Early tablet-based AR control systems for appliances in homes with preset interfaces and tags have been studied, but not widely adopted [14]. Some work has been done to create AR systems with more subtle or visually appealing markers [8, 19]. Recent work in visible light communication (VLC) enable data dissemination through modulation of LEDs, which can be captured on a smartphone camera, while remaining mostly imperceptible to the human eye [10, 20]. This can potentially act as a visual "landmark" for devices in MAR.

Localization. Recent work has demonstrated integration of novel indoor localization techniques in AR. ALPS makes use of ultrasound beacons that broadcast audio that can be picked up on smartphone microphones and used to accurately determine location and orientation of the user [11]. Similarly, use of ultra-wideband (UWB) can accurately locate tags that can be attached to phones or other objects [21]. The location information obtained from these techniques can be used to help render an accurate and persistent AR environment for all users and between multiple uses [12, 13].

3 APPLICATIONS

In the MAR browsing model, a user would be able to point their phone at the device of interest, and immediately discover the interface for it. The interface would link to a web page which can make use of a cloud service, a local network protocol, or a browser-provided JavaScript Bluetooth API to interact with the device. We explore a few applications for which an MAR-based browsing model might be useful.

3.1 Smart Home Devices

The smart home has been the primary focus for the consumer IoT market. Home apps, like Apple Home and Google Home, provide a single console interface for all the smart devices in a house. When a moderate number of compatible devices are present, it is often difficult to remember which interface corresponds with which device, even when each one is appropriately labeled on the console, and used regularly. Additionally, to be compatible with such consoles, devices must utilize specific proprietary protocols, and are limited in the type of interface they can display on the console. For this reason, many IoT manufacturers struggle to or opt not to integrate with these consoles. Regardless of compatibility, each device still typically requires installation of its own mobile app to function.

With the MAR browser, finding the interface for a device would be similar to finding one in the physical world. Instead of the Home apps' strategy of relying on the user's recall of a particular device, the MAR browser would utilize the user's recognition by allowing the user to find the appropriate interface by pointing to the device of interest, much like inspecting a lamp to find a switch.

3.2 Device Setup

To ease the process of connecting devices to the Internet in a home, the MAR browser could potentially enable a quicker and more convenient method for device setup and configuration. Instead of requiring the user to download an app and to go through a setup process of entering ID numbers or accessing settings to initiate a connection, the device's interface can be opened in the browser and it can use visual verification methods and a direct connection via Bluetooth to sync with a particular device. Additionally, users can place target locations for device interfaces in the augmented reality space during this process.

3.3 Ephemeral Devices

Breaking out of the confined scope of the smart home, an MAR browser could potentially be utilized to enable new opportunities for quick, ephemeral interaction with new and easily-discoverable interfaces in the global Internet of Things. Interactive devices and interfaces could be deployed in public (e.g. stores, sidewalks, conference rooms, exhibits). Users would explore these spaces by effectively using the browser as an MAR "window" into the public web. Support for device discovery and ephemeral interaction of this kind would be a major departure from the app-per-device and single-scoping models that are prevalent in IoT. Enabling this, however, will require important consideration of the methods for retrieval of data about devices in public and both the virtual and physical association of the appropriate interfaces to those systems.

4 OVERVIEW

To begin developing an AR-based browsing architecture, we consider design decisions about platform, target identification, user interfaces, and scope. To help aid the understanding of requirements and limitations, a prototype implementation, shown in Figure 2, has been created. This demonstrates one mechanism for AR-based discovery and interaction in a known environment, and helps inform some of the design decisions that would need to be considered.

4.1 Platform

The most straightforward implementation of the MAR browser requires an Internet-connected mobile phone or tablet that has sufficient hardware and processing support for augmented reality. Most flagship smartphones released in the last 4 years are capable of this at least with third-party software. More recent iOS (iPhone 6s, 7, 8, X, XS, XR, SE; iPad 5th Gen, Pro) and Android (Google Pixel 1, 2, 3; Samsung Galaxy S7, S8, S9, Note8, Note9) platforms have official support for MAR with dedicated software development kits from Apple and Google. Making this a mobile-based system, rather than one based on a head-worn device or a webcam, makes it more accessible to a larger population of current users and the touch interface can make it intuitive to use. Our implementation is built on iOS using ARKit.

4.2 Targets

One of the more challenging parts of this system is defining how smart devices are identified as "targets"—locations which, when viewed on the phone's camera, prompt the phone to overlay associated content (e.g. images, videos, or, in our case, a linked fully-functioning user interface for the device).

One of the common ways to do this is with image recognition. If the phone's AR browser is trained to recognize the image of the device, it can overlay the interface relatively easily. The prototype is an implementation of such a system, as depicted in Figure 2. In this example, the target is identified visually using a trained image of the lamp. The challenge with this is one of scalability. It is unlikely that the browser can be trained to recognize every device in every angle and lighting scenario.

Alternatively, the geolocation (GPS coordinates) of devices can be obtained during the setup of the device. This, however, would likely only apply to stationary devices and would not account for mobile or wearable devices. Additionally, geolocation on phones is still quite coarse-grained (m-level accuracy) and development of a global map of devices may prove to be an intractable problem.

More recent phones are capable of creating their own local mapping of targets in their environment. It is possible for local mappings to be shared from phone to phone, but the variance in data and lack of a common origin point of reference may render the data useless when trying to load it on a different phone or even reload it on the same phone at a different time. A possible method to improve accuracy with local mappings might be to have users initially place their devices on a designated origin when they enter an IoT-outfitted space. This could re-calibrate the device's orientation to allow the phone to more accurately place known targets in a space, and to enable persistence of location between uses. One could also envision using a mount outfitted with NFC (near-field communication) or

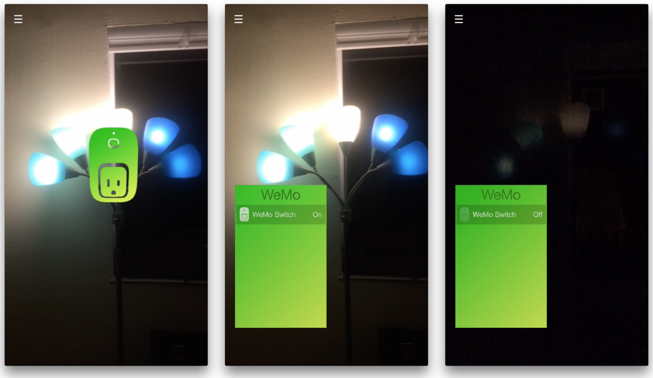

Figure 2: Screenshots of a demo implementation of a mobile augmented reality browser. In this proof-of-concept implementation, the smartphone camera identifies the target and displays an indicator (the favicon for the target's URL) which the user may touch to open the device's linked web interface. Once opened, the interface, an HTML web page, uses JavaScript and a local network protocol to allow the user to interact with the Wi-Fi-connected device, and toggle the light on or off.

Bluetooth as the designated origin, which could provide a URL to load a space's map on to the phone. This way, any smartphone or tablet entering the space can retrieve the appropriate local mapping instead of needing to rely on global geolocation.

The devices themselves can broadcast their presence to the phone over the local Wi-Fi network or Bluetooth Low Energy. Like with Google's Physical Web, the devices can provide a URL in their broadcast advertisements that would both help to identify the device and point to an address where the device's user interface is served. This can also help to simplify how devices are found. While fine-grained location (cm-level accuracy) cannot be obtained from the detection of Wi-Fi or Bluetooth, it indicates the relative proximity of the device, which can 1) inform the user of its presence and 2) provide a more simplified, scoped search-space for the phone to identify targets.

For example, a light bulb can advertise `https://lightswit.ch/control`, the address of its user interface, over Bluetooth Low Energy. Once detected on a user's phone, the device can be listed on-screen as a nearby "thing". At the same time, the phone can also refine its search-space of targets to objects that look like light bulbs associated with the `lightswit.ch` URL or, perhaps, known geolocations of such devices. If the phone is unable to identify the device, but the user has knowledge of its location, he or she can potentially contribute a picture or an updated coordinate to help identify the device in the future.

4.3 Target Proxies

The reality in many environments is that the device itself may not be a desirable target. For instance, a smart power meter outlet may not be visible when in use since the plug for the appliance it is metering may obscure it. Additionally, the data is directly related to the metered appliance, and the user may more naturally associate the interface for the meter with the appliance rather than the meter itself. The user may prefer that the interface for the meter be placed closer to the actual appliance. Figure 2 is also an example of this. The lamp was a chosen as the target location for the interface rather than the smart switch device at the outlet, as it more closely relates to the object of the intended activity of interaction. Consideration for these situations provide greater reasoning for allowing users to define/modify the placement of interfaces for the devices they own, or utilizing standard visual identifiers like QR codes that can be placed at the desired location.

4.4 Interfaces

Once a target has been identified, the browser should be able navigate to and display a linked user interface for the device that can display data from and interact with the device. To facilitate this in a general and scalable manner, standard web tools can be used to create the interface content. This means that the UIs are, in effect, web pages created with HTML, JavaScript, and CSS. This would allow developers to maintain their own creative liberties with the interaction model and interface aesthetic. The interface can make use of a browser-provided JavaScript Bluetooth API or a local networking protocol to enable local communication between the browser and the device. Alternatively the interface can continue the trend used by mobile apps of proxying commands through a cloud service.

In the case of the example shown in Figure 2, the interface is a web page that is served on the local Wi-Fi network. Whenever the user presses the switch displayed on the interface, it issues a server-side command to the network-connected smart switch device through a UPnP protocol.

4.5 Scope

Though it potentially limits a range of possibilities with public and ephemeral discovery, the browsing model can work as a local system that is focused on operating with devices in a user's home or work environment. Users can easily feed the system images of the devices in the actual environment they reside, providing targets that can be recognized more reliably. Alternatively, the user can define or correct an object's location using a drag and drop interface which will then be stored as coordinates in the phone's local mapping of the environment.

However, a system that is capable of working at a larger scale is more desirable. Identifying things as targets by image recognition is made more feasible when using the devices' Bluetooth or Wi-Fi broadcasts to refine the search-space. Use of standardized visual identifiers such as QR codes or AprilTags may improve performance, especially when devices are not stationary. Locations can be defined/corrected by the crowd using a drag-and-drop style interface to obtain new images of the target or modified geolocation.

Overall, defining and determining how the physical scope for interaction is interpreted will be important. Depending on the technologies used, a "nearby" device could be considered one within the device's Bluetooth range, on the device's Wi-Fi network, in identifiable visual range of the phone's camera, or within an arbitrary distance of the phone's geolocation.

5 RESEARCH QUESTIONS

In this section, we discuss some research challenges and questions that should be considered when designing and implementing the MAR browsing architecture.

5.1 Location Determination & Accuracy

A key consideration for the MAR browser is determining where interfaces are placed in augmented reality and ensuring that the placement corresponds with a user's intuitive understanding of how that interface relates to the physical environment. What are the methods to accomplish this? Can we do this without — or at least with minimal — user setup? Can we support both stationary and mobile objects of interest? How do we properly identify and distinguish interfaces for multiple targets in close proximity? How will different types and levels of lighting affect the accuracy of target identification and interface placement? We discussed some of these considerations in Sections 4.2 and 4.3. Some solutions for improvement might include updating location by crowd-sourcing user location (verification by proximity), storing device location meta-data in the cloud which can be downloaded based on user location, or "annotating" locations and devices of interest with QR codes that link to well-defined schema.

5.2 Communication Topology

Once a web-based user interface is opened, how do we enable direct interaction with the device? We suggest enabling support for local interaction using a JavaScript Bluetooth API for Bluetooth devices. Interfaces for Wi-Fi devices can implement local network protocols if the web content is served on the local network. Alternatively, as many mobile apps currently do, the interface can issue commands through a dedicated cloud service for devices with continuous Internet connectivity. This would effectively allow developers to create and deploy app-like UIs for their own devices.

5.3 Usability & User Experience

The intention of this browsing model is to begin taking steps to make software-based interaction with IoT devices feel as intuitive as physical interaction with our surrounding environment. Besides physical placement of interfaces, what are the design decisions that help interface feel more tangibly and physically tied to the devices they control? Can we make interaction via mobile as convenient as on-device interfaces? Can we give creators the power to create interfaces suitable for their product rather than constrain the types of services that are available? How do we design the browser to better promote ephemeral discovery and interaction in public?

Augmented reality, while intuitive in many ways, can still generally feel awkward to users. Will users adjust to the paradigm of placing a phone between them and the world around them? Interesting approaches have been taken in the past to help reduce awkwardness in MAR, like using camera attachments to facilitate a more comfortable screen angle [4] or incorporating novel methods of feedback [16].

While not entirely important at least with initial prototypes, consideration of occlusion and other visual features that help provide a sense of realism to the virtual objects in physical space will eventually be useful for improving the user experience.

5.4 Developer Experience

Ideally, the process of creating devices that are compatible with the browser and developing interfaces that are discoverable, operable, and associated on the browser should be a relatively convenient process compared to the current siloed mobile app approach. What should the programming environment look like and what tools should be made available to developers?

Because the browser would essentially run on an extension of web standards, developers could still make use of a typical web programming environment to create the interfaces. Some browser-specific APIs will likely need to be defined to help facilitate functions like Bluetooth communication and establish some camera and visual verification protocols. Determining the exact set of functions that is required will be important. For device development, standard advertisement services over Bluetooth or Wi-Fi (e.g. mDNS) can be used to broadcast URL and other metadata to the phone. The developer can also set up a server to host a UI on the device itself.

5.5 Browsing Model

The UIs displayed on a user's smartphone can be physically tied to the devices they control or to a relevant point in physical space. But does this present a larger benefit than previous paradigms, like simple linking through Physical Web or QR codes alone? While Physical Web provides a convenient means for content discovery, it can either overwhelm the user with notifications for a long list of web pages or prevent relevant results from being presented in an attempt to filter that list. Additionally, it could be difficult for the user to determine which interface is associated with which device.

QR-linking does enable more physically-tied associations of content to target objects. However, at present, the common uses for QR seem to be very deliberate, single-time actions like linking to a download for an app or to a configuration web page within a browser. It also requires switching back and forth between camera and browser to open multiple interfaces. The MAR browser would effectively be combining camera, browser, and local network protocols to allow for a more seamless experience when inspecting and interacting with a space.

Once a model for linking interactive web interfaces to target devices is established, are there more interesting and novel features that could be supported through the MAR browser? Perhaps, for instance, a method could be established to allow users to easily perform trigger-action programming connecting multiple devices in their environment.

5.6 Data & Privacy

Because the system uses camera, location, and data about devices in local and private environments, there are risks associated with the size and sensitivity of the data. It is important to determine exactly what data is necessary to save locally or send online, and develop measures to ensure that no more than the necessary amount is retained, transmitted, or shared. Additionally, the browser model will likely require some defined notions of ownership and permission, and rules to enforce them. Safeguards will also need to be put in place to detect and prevent links to malicious web interfaces. It is essential that, at least, the standard practices of web security are preserved and supported through the browser.

5.7 Power Usage

Currently, use of augmented reality is a notable drain on phone batteries, which may discourage people from using the browser, especially in public, where phone charging is often limited. Phone companies seem to be allocating a considerable amount of effort into making it more efficient. Considering what practices make more efficient use of energy will be useful when implementing the design. For example, using flat imagery like favicons and rectangular web page interfaces in MAR will generally require less energy than rendering 3D objects and controls.

6 CONCLUSIONS

The technology to support mobile augmented reality exists, and there is high interest and effort put into continuing to improve it. Creation of an MAR-based browsing architecture would primarily consist of putting the right pieces together and making concrete decisions on standards, particularly for target recognition and device discovery. Further studies will be required to better assess and improve the usability of a such a system, but utilization of this architecture may enable us to begin taking steps to break free of the walled-garden infrastructure that is stifling the Internet of Things, while also providing compelling use cases for the augmented reality technology that is becoming more prevalent on our mobile devices.

7 ACKNOWLEDGMENTS

We wish to thank our shepherd, Xia Zhou, and the anonymous reviewers for their detailed comments and feedback. This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. This material is also based upon work partially supported by the National Science Foundation under grant CNS-1824277, as well as by the NSF/Intel Partnership on CPS Security and Privacy under grant CNS-1822332.

REFERENCES

[1] Apple. 2018. Apple Developer Documentation. (Oct 2018). https://developer.apple.com/documentation/

[2] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, I. Lopez, R. Ramamoorthi, S. Sheorey, S. White, and L. Zhang. 2008. Searching the World's Herbaria: A System for Visual Identification of Plant Species. 116–129. DOI: https://doi.org/10.1007/978-3-540-88693-8_9

[3] B. Brumitt and J. Cadiz. 2001. " Let There Be Light": Examining Interfaces for Homes of the Future.. In *INTERACT*, Vol. 1. 375–382.

[4] A. Colley, W. Van Vlaenderen, J. Schöning, and J. Häkkilä. 2016. Changing the Camera-to-screen Angle to Improve AR Browser Usage. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 442–452. DOI: https://doi.org/10.1145/2935334.2935384

[5] G. Furnas, T. Landauer, L. Gomez, and S. Dumais. 1987. The Vocabulary Problem in Human-system Communication. *Commun. ACM* 30, 11 (Nov. 1987), 964–971. DOI: https://doi.org/10.1145/32206.32212

[6] Google. 2017. The Physical Web. (Jun 2017). https://google.github.io/physical-web/

[7] Google. 2018. Google Developers. (Jul 2018). https://developers.google.com/products/

[8] V. Heun, S. Kasahara, and P. Maes. 2013. Smarter Objects: Using AR Technology to Program Physical Objects and Their Interactions. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 961–966. DOI: https://doi.org/10.1145/2468356.2468528

[9] N. Kumar, P. Belhumeur, A. Biswas, D. Jacobs, W. Kress, I. Lopez, and J. Soares. 2012. Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In *Computer Vision – ECCV 2012*. Springer Berlin Heidelberg, Berlin, Heidelberg, 502–516.

[10] Y. Kuo, P. Pannuto, K. Hsiao, and P. Dutta. 2014. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom '14)*. ACM, New York, NY, USA, 447–458. DOI: https://doi.org/10.1145/2639108.2639109

[11] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe. 2015. ALPS: A Bluetooth and Ultrasound Platform for Mapping and Localization. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*. ACM, New York, NY, USA, 73–84. DOI: https://doi.org/10.1145/2809695.2809727

[12] P. Lazik, A. Rowe, and N. Wilkerson. 2018. ALPS: The Acoustic Location Processing System. https://www.microsoft.com/en-us/research/uploads/prod/2017/12/Patrick_Lazik_2018.pdf. (2018).

[13] P. Lazik, A. Rowe, and N. Wilkerson. 2018. Realty and Reality: Where Location Matters. https://www.microsoft.com/en-us/research/uploads/prod/2017/12/John_Miller_2018.pdf. (2018).

[14] J. Lee, J. Kim, J. Kim, and J. Kwak. 2007. A Unified Remote Console Based on Augmented Reality in a Home Network Environment. In *2007 Digest of Technical Papers International Conference on Consumer Electronics*. 1–2. DOI: https://doi.org/10.1109/ICCE.2007.341516

[15] H. Ling. 2017. Augmented Reality in Reality. *IEEE MultiMedia* 24, 3 (2017), 10–15. DOI: https://doi.org/10.1109/MMUL.2017.3051517

[16] C. Liu, S. Huot, J. Diehl, W. Mackay, and M. Beaudouin-Lafon. 2012. Evaluating the Benefits of Real-time Feedback in Mobile Augmented Reality with Hand-held Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2973–2976. DOI: https://doi.org/10.1145/2207676.2208706

[17] D. Marques and R. Costello. 2015. Skin & bones: an artistic repair of a science exhibition by a mobile app. *MIDAS. Museus e estudos interdisciplinares* 5 (2015).

[18] Hewlett Packard. 2018. HP Reveal. (Mar 2018). https://www.hpreveal.com/

[19] J. Platonov, H. Heibel, P. Meier, and B. Grollmann. 2006. A mobile markerless AR system for maintenance and repair. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*. 105–108. DOI: https://doi.org/10.1109/ISMAR.2006.297800

[20] N. Rajagopal, P. Lazik, and A. Rowe. 2014. Visual light landmarks for mobile devices. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 249–260. DOI: https://doi.org/10.1109/IPSN.2014.6846757

[21] N. Rajagopal, J. Miller, K. Kumar, A. Luong, and A. Rowe. 2018. Welcome to My World: Demystifying Multi-user AR with the Cloud: Demo Abstract. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '18)*. IEEE Press, Piscataway, NJ, USA, 146–147. DOI: https://doi.org/10.1109/IPSN.2018.00036

[22] D. Van Krevelen and R. Poelman. 2010. A Survey of Augmented Reality Technologies, Applications and Limitations. *International Journal of Virtual Reality* 9, 2 (June 2010), 1–20. http://www.ijvr.org/issues/issue2-2010/paper1%20.pdf