WiFi proximity detection in mobile web applications

Clemens Nylandsted Klokmose[1,2] clemens@cs.au.dk
Matthias Korn[3,1,2] korn@iupui.edu
Henrik Blunck[2] blunck@cs.au.dk

[1]Center for Participatory IT, Aarhus University, DK-8200 Aarhus N, Denmark
[2]Department of Computer Science, Aarhus University, DK-8200 Aarhus N, Denmark
[3]School of Informatics and Computing, Indiana University, IUPUI, Indianapolis, IN 46202, USA

ABSTRACT
We present a technique for enabling WiFi proximity detection in mobile web applications based on proximity-adaptive HTTP responses (PAHR). The technique requires zero installation on the client and is client platform independent. Our reference implementation ProxiMagic is low-cost and provides robust and responsive interactivity based on proximity detection. We demonstrate the technique's applicability through a real-world example application deployed during a month-long participatory art exhibition. We document the reliability and suitability of the simple proximity detection employed in ProxiMagic through a controlled experiment.

Author Keywords
WiFi proximity detection; context-awareness; HTTP; mobile web applications; mobile devices; zero installation.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g., HCI): Miscellaneous

INTRODUCTION
In our research we study how IT can enable participation in activities in and around a local space. In a recent interdisciplinary research project we explored how we could let visitors of an art exhibition participate in the curatorial activity of describing and interpreting the exhibited artworks using their personal mobile devices [11]. A central goal in the project was to minimize the effort required for the occasional visitor to participate in the collaborative writing and reading activity. This lead to four design requirements: (1) Visitors should be able to edit the curatorial text of an artwork they are standing in front of from their personal device with minimal or no navigation required on their part—i.e., navigating from one artwork to another should happen *automagically* as one moves around in the physical space of the gallery. Hence, the system should reliably detect when a user device is within 2-4 meters of an artwork. (2) Because visitors may only ever visit the exhibition once, they should be able to participate without having to install any software on their personal devices and requirements for bootstrapping (e.g., configuring the device) should be minimal. (3) Visitors with a wide variety of devices and operating systems should be able to participate. (4) The needed hardware infrastructure in the space should be cheap and based on off-the-shelf components to fit a meager institutional budget. We believe that these requirements of zero install, minimal navigation, platform independence, and low cost apply not only to art exhibitions, but to various kinds of public and semi-public spaces that are sought to be augmented with an interactive digital layer accessed from personal devices.

We realize these requirements through our novel technique based on *proximity-adaptive HTTP responses* (PAHR) and present our reference implementation *ProxiMagic*. In brief, the technique adapts the response to an HTTP request to a local web server based on clients' proximity to points of interest in the local space. These points of interest are instrumented with sensing nodes that continuously report proximity data to the web server. Hence, the client needs nothing but a wireless network interface and a standard web browser.

We report insights from one of only few real-world uses of WiFi positioning [7] as a means for engineering novel interactive systems in physical spaces. While the literature on WiFi positioning has high-accuracy absolute positioning as the ultimate goal [10], we demonstrate that simple, low accuracy proximity detection is adequate for building a novel interactive system.

EICS 2014, June 17–20, 2014, Rome, Italy.

http://dx.doi.org/10.1145/2607023.2610281

RELATED WORK
Location-awareness has been part of the early ubiquitous computing vision and prototypes, where it commonly requires custom software (or even hardware) on the client side. The ParcTab system was designed around an IR-based local area network that could identify the room the user was in and deliver relevant content in the user interface [15]. The Cyberguide, aiming at a design for a virtual tour guide, combined indoor IR-based proximity detection with outdoor GPS tracking to provide access to services and content related to a location [2]. Both the ParcTab and Cyberguide relied on custom client software. GUIDE was similarly developed as an intelligent tourist guide [5]. It provided outdoor proximity detection not through GPS but with the help of several WiFi base stations. GUIDE in fact employed location-aware web content delivery, however, using a customized web-browser capable of navigating web content based on proximity data from the WiFi base stations. Today, outdoor positioning has become a standard feature of modern mobile devices, and the W3C Geolocation API [12] allows access to location information in the web browser. They do not, however, support indoor positioning.

Apple's Bluetooth Low Energy (BLE)-based[1] iBeacon[2] is a promising technology for proximity detection. iBeacon relies on low-cost signal emitting beacons, that can be placed at points of interest in a local space such as a museum or a mall. The Proximity Profile of Bluetooth 4.0 (PXP 1.0)[3] allows one device to detect whether another device is within a close physical range. As in our approach, physical proximity is estimated using the radio receiver's RSSI value, which has no absolute calibration of distances. The iBeacon technology is currently only available from within an (iOS 7 or Android 4.3) app on a device supporting Bluetooth 4.0. Hence, (as of now) it does neither meet our requirement of platform-independence nor of zero install on the client device. Moreover, there is no mention yet of enabling access to the iBeacon API from the web browser. Lastly, concerning backward compatibility, iBeacon does not support devices without Bluetooth 4.0, whereas our approach transparently works with any WiFi-enabled device with a web browser.

There exists a multitude of indoor positioning techniques that provide absolute positioning to varying degrees of accuracy. These techniques are based on, e.g., infrared [14], ultrasound [16], WiFi fingerprinting [3], Bluetooth [1], GSM fingerprinting [13], or even GPS [6]. Proximity to points of interest (POIs) can be inferred as an intermediate result from WiFi fingerprinting, even if sensing nodes do not align with the POIs [9]. However, fingerprinting requires a costly initial collection phase in order to establish a database of signal strength readings for later comparison; furthermore, this phase has to be repeated whenever the environment or the sensing node setup changes [3, 8]. In contrast, our technique relies on instrumenting all POIs with cheap sensing nodes. This strategy adds to the physical deployment complexity, but provides good proximity-detection accuracy even without fingerprinting—and thus avoids costly calibration or collection procedures.

TECHNIQUE

In order to provide *proximity-adaptive HTTP responses* (PAHR), a web server looks up the proximity of requesting devices on a given wireless network to a number of points-of-interest, and adapts the responses accordingly. PAHR thus enables to, e.g., serve information about the POI that a device is closest to (e.g., the closest artwork in our art exhibition case). PAHR requires that the web server can map the client IP in the header of the HTTP request to information about the proximity of the device to the relevant POIs. Based on this proximity information, PAHR then allows to either switch between different static web pages, or to dynamically update the content of a webpage through AJAX.

In our realization of PAHR, we instrument POIs with dedicated sensing nodes, (low-cost) credit card-sized single-board computers equipped with a WiFi adapter running in passive mode. 802.11 packet capturing on a given WiFi network is employed on the sensing nodes to detect the presence of wireless devices and their communication with wireless access points.

Based on the packet capturing (filtered by network), the sensing nodes continuously report the *received signal strength indicator* (RSSI) from each of the detected devices to the web server as tuples in the form of: `{sensingNodeID: int, readings: list({clientMAC: string, rssi: int})}`. The web server stores a mapping of the MAC addresses of all (detected) wireless devices on the network and their measured signal strengths at all the sensing nodes. This mapping is used when a device on the local network makes an HTTP request to the web server: The web server can extract the client IP from the HTTP request header, and through an address resolution protocol request on the local network (using `arp` on Unix-like operating systems) map it to the MAC address of the device. Based on a configured global signal strength threshold that signifies being in proximity, the server then produces a list of received signal strengths at all sensing nodes to the given device and uses this information to adapt its response. This means that the web server can provide, e.g., a basic HTTP API to retrieve the list of proximity information for a client to all points-of-interest, or the server can simply redirect HTTP requests based on proximity.

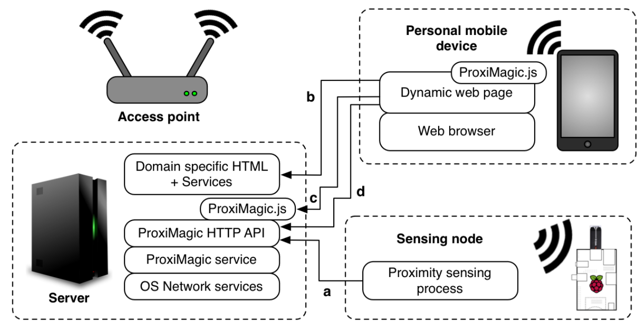

Figure 1. Components of ProxiMagic.

REFERENCE IMPLEMENTATION

ProxiMagic[4] is our reference implementation of a PAHR-based system. Figure 1 shows the interaction between the components involved: **a)** The sensing node continuously posts proximity data to the web service. **b)** A client loads the dynamic web page which **c)** also loads the client side ProxiMagic API *ProxiMagic.js*. **d)** Through the ProxiMagic client side javascript library, the dynamic web page requests the nearest objects of interests, resulting in an operating system request on the web server to map the IP of the client to a MAC address, and finally returning the proximity information to the client.

A requirement for ProxiMagic was that the hardware should be *low-cost* and off-the-shelf. Hence, the sensing nodes were built using the US$35 single-board computer Raspberry Pi (model B) running Raspian (Wheezy release). The Raspberry Pi were equipped with a ~US$15 D-Link DWA-140 (Rev b3) USB wireless network card. For monitoring the wireless network we used the open source packet capturing tool *Airodump-ng* (ver. 1.1).[5] While not interested in the actual data transmitted on the wireless network, we were interested in knowing the origin and destination of a given packet which we could extract using Airodump-ng. The sensing nodes ran a small Java process parsing and filtering the output of Airodump-ng, and posting it to the ProxiMagic web service.

[1] `http://www.bluetooth.com/Pages/low-energy.aspx`
[2] `http://support.apple.com/kb/HT6048`
[3] `https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=239392`

[4] Download at `http://proximagic.projects.cavi.au.dk/`
[5] `http://www.aircrack-ng.org/doku.php?id=airodump-ng`

The ProxiMagic web service was implemented in PHP on an Apache web server running on Ubuntu Linux. We used the *arp* implementation provided by Ubuntu to perform the MAC-to-IP address lookup. The web service provided a simple HTTP API for the sensing nodes to post their data, and for web clients to retrieve proximity information to points-of-interest. The web service additionally served a client side Javascript library enabling the client to subscribe to proximity events (by polling the web service). The server can furthermore host and serve application-specific web content, which it did in our example application described below. However, the content does not have to be served from the same machine. On the client side the proximity information can, e.g., be used to dynamically load new content in an iFrame based on proximity events (which is what we did).

STRENGTHS AND LIMITATIONS

The strength and novelty of PAHR is the provision of proximity-based services i) on all WiFi-enabled devices capable of communicating through the HTTP protocol, and ii) without the need to install non-standard software on client devices.

Since we do not rely on an installed app, we have little control over the wireless network card on the device—i.e., we cannot keep the WiFi card from powering down on inactivity. As a result, we can only obtain proximity information when the device is generating network traffic. Therefore, the proposed technique can be considered to be *on demand*, and would not support, e.g., push notifications of proximity when the device is sleeping or otherwise not communicating (as promised, e.g., by Apple's iBeacon).

PAHR relies on the server being able to maintain a mapping between the IP addresses of requesting devices and their MAC addresses. In our reference implementation this requires the server to be on the same IP subnet as the clients for it to make the address resolution protocol request. In a more complex network infrastructure, this problem could be overcome by delegating ARP requests to the sensing nodes. However, this strategy would require at least one sensing node on each IP subnet of the network where there are points of interest. The ProxiMagic implementation is currently limited to a wireless network comprised of a single access point. It is possible to support larger wireless networks with multiple access points on different channels. This would, however, increase the latency of the proximity detection significantly as the sensing nodes would have to hop between channels. While in our deployments the sensing nodes have been connected to the wired network for communication with the web server, we have successfully tested wireless sensing nodes being both connected and monitoring the same wireless network with a single WiFi adapter. However, this limits the possibility for channel hopping even further.

ProxiMagic requires that each POI is instrumented with a sensing node. In the light of the recently announced iBeacons, this seems to be an approach adopted by industry as well. It is furthermore important to note that our proximity detection is based on received signal strength indicators (RSSI). Hence, it is only an approximate value and cannot be mapped to a metric distance without more sophisticated triangulation between sensing nodes. In effect, this means that a threshold RSSI indicating proximity (as we use) will result in different actual distances from device to device and other environmental conditions.

REAL-WORLD APPLICATION EXAMPLE

ProxiMagic was developed and deployed as part of the system Local Area Artworks (LAA) in a month-long participatory art exhibition in a contemporary art gallery in Denmark. LAA was designed to engage visitors of the exhibition in collaboratively writing descriptions and interpretations of displayed artworks using their own personal devices. Description panels next to artworks with text traditionally written by curators were replaced with digital panels in the form of framed iPads, and visitors could collaboratively edit the texts on the panels from their personal devices if they were in proximity of the artworks.

LAA was an interdisciplinary project. During the exhibition the use of the system was studied in the wild through observations, interviews, and logging (initial findings can be found in [11]). Throughout the deployment period of LAA, 141 unique devices were tracked by our sensing nodes, and 118 (84%) also communicated with the web server. On average, 5.4 unique devices interacted with the system per day with a peak of 34 different devices on the opening night. Three iPod Touch were available for borrowing at the cash desk, and these were responsible for 44% of the logged activity in the system. One of our goals with LAA was to minimize the burden put on the *occasional* visitor in the form of avoiding installation on personal devices and complex navigation tasks. Hence our requirements for zero installation and 'automagic' navigation.

ProxiMagic was used to let visitors interact with specific artworks without the need to manually connect to them (e.g., through scanning QR codes) or to select them in the interface. Instead, people could navigate the gallery space with their feet and ProxiMagic would take care to automatically show the artwork they were closest to. In addition to the ProxiMagic base layer, LAA contained an application layer to enable location-based collaborative writing about specific artworks in the gallery space. The application layer was based on a modified EtherPad[6] installation, which is a web-based collaborative writing engine. An EtherPad view was used both on the personal devices and the digital panels. This enabled that the text on the digital panels updated live when the visitors edited it on their devices, which emphasized the co-located use. Furthermore, the digital panels displayed a row of dots indicating how many devices were in proximity of it and how many of those were actively editing (colored if actively editing, grey if not).

Visitors connected their own personal device (or a borrowed device) to an open wireless network. This wireless network did not provide Internet access, and all HTTP requests were redirected to our web server on the local network. Hence, when visitors opened a URL in a browser on their phone, they would automatically be redirected to our web-based system. If they moved into proximity of a panel (within 2 to 4 meters), ProxiMagic would redirect the browser to an editable version of the text for the particular artwork.

Members of the audience could only participate actively by being there, by being in close physical proximity of an artwork. Once they moved out of proximity of the art piece, they were no longer able to edit the texts. Moving to another artwork automatically redirected to the respective editable text. Moving out of proximity

[6] `http://etherpad.org`

of any panel, visitors were presented with a floor plan of the art gallery indicating the locations of the panels.

Deployment Experiences. To our surprise, the visitors we observed and interviewed did not explicitly question nor, even when directly asked, reflect upon in which way the artworks were 'served' to them. For the users, it seemed natural to move around the space and thereby be navigated to different artworks on your personal device; it just worked. However, they did experience some ambiguities in our proximity detection that fundamentally underlie all radio-based approaches [4]. We found that people employed their own strategies to work around and compensate for the way proximity was defined and detected. Due to our proximity being an approximate measure, users did not know when they were within proximity at any given point (i.e., in which zones they could move around freely). Hence, they tested this out themselves by moving back and forth and observing changes on the interface. For example, users adapted when they did not get the content they expected by moving around or by holding their phones closer to the panel. Some visitors had to go really close with their device to a given point-of-interest, almost touching it. As our experiment below shows, varying power of the WiFi radios across devices, the users' orientation and their grasp of the device are to blame.

While our approach was zero install, it was not zero setup. Visitors were required to connect to a specific wireless network and navigate to an arbitrary page in the browser. The latter seemed hard to grasp, hence we ended up instructing visitors to navigate to a fictional URL. Not all visitors were comfortable with changing network settings on their devices, and hence required assistance from the staff. However, even non-tech savvy visitors grasped the logic of having to connect to the local wireless network of the exhibition space to participate in the discussions from their personal device.

EVALUATION OF PROXIMITY DETECTION

In the following, we demonstrate that the proposed simplistic technique for proximity detection reliably detects devices within a 2-4 meter radius of a POI—across a wide variety of mobile user devices and conditions, per our design requirements.

Setup. We conducted a series of 13 controlled experiments in a small sports hall (16x16 meters) with one sensing node and one user moving for ca. 3 minutes while carrying a mobile device. For the experiments we used variants of the following *default conditions*: using an iPod Touch running iOS 6.1.3 as user device, grasping it naturally, always facing the sensing node, and stop-walking (i.e., step—2s stop—step) along a fixed path that is visualized in Figure 2 (left). Each of the 12 experiment variants deviated from the default conditions in exactly one of the following 4 aspects:

- user device of various form factors and platforms: a Samsung Galaxy Nexus GT-i9250 running Android 4.3; an iPad 3 running iOS 7.0.2; and a late 2010 MacBook Air running Mac OS X 10.8.5
- orientation of the user: always facing away; facing right; and facing left
- the device-holding type: grasping with two fingers only; shielding with two hands; in-hand, arm pointing downward; and in pocket
- moving freely in the hall: walking; and running

As sensing node we used a Raspberry Pi, equipped with a D-Link DWA-140 WiFi antenna, placed at 2m distance from the center of the hall's south wall. The sensing node posted signal strengths from user devices at ca. 10Hz to a web server for data logging. We obtained ground truth of the user's location, and thus of her distance to the sensing node, via a floor-mounted Leuze ROD4-50 Plus laser scanner that was co-located with the sensing node. The scanner scanned in an angle space of 180 degrees for objects near the floor (e.g., feet) with an angular resolution of 0.35 degrees and a frequency of 10Hz. Each reading posted to the server from a sensing node was logged with the last reading from the laser scanner. Additionally, video of the experiments was recorded.

ProxiMagic is designed to detect proximity events, i.e., to detect when the user comes closer (or moves further away) than a prescribed distance. This proximity zone is associated with a pre-configured received signal strength indicator (RSSI) threshold. This threshold is experimentally established in order to approximate a metric proximity zone, e.g., of 4 meter radius around the sensing node. In the following, we evaluate ProxiMagic's accuracy for *several* proximity radii in order to gain statistical insights. To be able to do so, we produced a complete mapping m between between distance and signal strength. We chose to derive this mapping via using a simple regression scheme and solely on the basis of data from the default condition experiment. We decided against using more experimental data—and thus against learning a (potentially better fitting) mapping—foremost in order to evaluate and illustrate two features of the proposed technique: i) that it does not require time-consuming calibration procedures, and ii) that it provides reasonable results even when the real-world use conditions vary in terms of devices, grasp, and orientation.

Results. For illustration purposes, Figure 2 (left) plots the signal strength measurements for the stop-walk experiment, as obtained by the Pi for the default condition at their respective ground truth locations. Complementing, Figure 2 (right) illustrates the distances measured by the laser scanner (blue) compared to the distances as obtained through the mapping m of the measured signal strengths (green). This mapping thus allows us to systematically compute (and to statistically analyze) the recorded *distance errors* coming with the RSSI-estimated distances, i.e., how much these differ from the actual ground truth distances in our experiments. As an example, the figure furthermore illustrates, in orange, the RSSI-based estimates of whether or not the user is within a 4m proximity zone. It can be seen that the signal strength fluctuations can cause a local oscillation of the proximity detection result. In order to fight such 'flickering', smoothing techniques can be applied—at the expense, though, of a delayed detection: To obtain the smoother detection in Figure 2 (red), simple Kalman-filtering was applied.

Table 1 lists observed distance errors, averaged over each experiment's measurements. The overall error levels of, e.g., 2.64m on average across the four used devices, are higher than those usually provided by dedicated and more costly positioning systems. Nonetheless, the results suggest that ProxiMagic is suitable for its intended purpose—as illustrated by Figure 2 (right): The absolute error of 1.95m translates here to a responsive and almost always accurate proximity detection. Errors are also given differentiated

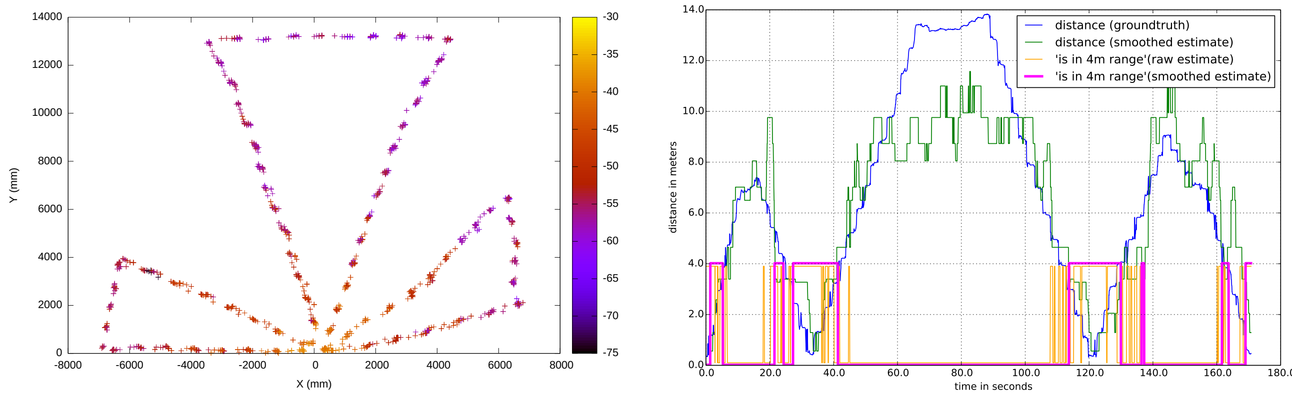

Figure 2. Measurements and results from the experiment with default conditions over time. Left: Ground truth in mm along X and Y, and signal strength on a color scale. Right: The distance to the user device, the signal-strength-based estimate of it; and whether the user is estimated to be in a 4m radius around the Pi.

	default	user device			user orientation			grasp				free-form	
		Galaxy Nexus	iPad	MB Air	facing away	facing right	facing left	minimal grasp	2-hand shielding	pointing down	in-pocket	running	walking
distance error (m)	1.95	2.69	2.81	3.14	3.62	2.50	2.39	2.57	5.23	2.54	2.93	2.07	2.72
distance error at ~2m	1.34	1.32	1.12	1.68	5.52	2.53	1.53	2.20	7.46	1.25	3.16	1.94	3.01
distance error at ~4m	1.72	1.77	1.68	2.19	4.57	1.92	1.83	2.92	6.77	1.81	2.64	2.00	2.96
distance error at ~6m	1.72	1.97	2.37	2.34	3.28	1.82	1.95	2.56	5.69	2.52	2.33	2.18	2.74

Table 1. Accuracy results for default conditions (grasping an iPod naturally, while constantly facing the wall with the Pi, and moving along a default path) and variations of it. Errors are given for all measurements, and specifically for those obtained at ranges around 2, 4, and 6m, respectively.

for individual user proximity ranges, i.e., for when the user's actual distance is close to (i.e., between $2/3$ and $3/2$ of) a proximity threshold of 2, 4, and 6 meter, respectively. The absolute error levels are similar across all of these three ranges; this implies that the relative error (i.e., relative to the user's actual distance) is higher for shorter distances. On this basis, we argue that the accuracy levels obtained can be deemed acceptable for use scenarios requiring coarser proximity thresholds, e.g., for our art exhibition case. However, the accuracy levels obtained for close-range proximity prohibit precise detection of fine-grained proximity changes within arm's length, e.g., for supporting gestures.

Table 1 furthermore provides evidence that the proposed technique provides generalizable results—and that it does not require a tuning of the technique to the wide range of device types or device grasps: The error levels for all four devices used, and also for the investigated device grasps are similar. An exception from this are the high error levels for the condition of the '2-hand shielding' grasp of the device—which lead to a overestimation of the user's distance. A similar shielding effect is observable for when the user is facing away from the sensing node. In contrast to the hand-shielding, a facing away can be expected in real-world use scenarios. Here, the overestimation though aids the proposed technique, since it prevents it from reporting proximity events when facing away, that is, when the user is most likely not actually approaching the POI in question.

Potential accuracy improvements. The results listed in Table 1 were obtained by smoothing signal strength measurements over $\Delta t = 0.5s$—which reduced distance errors by on average 15%. As discussed using Figure 2, a larger time window Δt leads to delayed proximity event reporting. Thus, a trade-off between distance error and responsiveness has to be made in dependence of the use scenario, and on the proximity distances relevant to it. While for larger distances the distance error may be more crucial, we argue that responsiveness becomes more critical for near proximity use cases: e.g., if immediate and fluid interactions with the approached object of interest should be supported. Another potential means for improving proximity detection accuracy is to observe for a time window Δt not only the average signal strength, but also its trend—indicating whether the user is approaching and/or turning towards a sensing node. It remains to be evaluated how the results given here generalize to other, e.g., larger or more secluded environments and to more complex WiFi setups. To this end, we produced results as in Table 1 using an alternative signal strength to distance mapping m' with data from another environment (a large office building complex) and using other devices; the obtained errors were only insignificantly higher on average than those given in Table 1. Finally, the accuracy gains achievable when utilizing costlier sensing nodes, specifically costlier antennas remain to be explored.

DISCUSSION

The experiment and deployment are limited to what extent they demonstrate scalability in terms of significantly more users or sensing nodes. There are a number of parameters impacting performance that could be evaluated experimentally such as the density of sensing nodes, their placement in relation to the layout of the room, the maximum amount of detectable wireless transmitters, and the density of people in a room. However, our experimental evaluation and our experiences from the real-world deployment show that it is feasible to use WiFi proximity detection as an interaction technique.

The opaqueness of the actual proximity distance to the users turned out to be an interesting property in the LAA deployment. Given the nature of our technique, we could not make a direct metric definition of proximity, e.g., draw a line on the floor where our defined zone of proximity would be. In effect, this meant that our users could not build up expectations regarding accuracy and rather tested in a playful manner how the proximity detection worked for a given art piece and their particular device. If proximity zones are, however, defined at a very close distance to support close-range interaction (e.g., under 1m), then the accuracy of the proximity detection becomes more important.

Since our technique is network-based (i.e., proximity detection is handled exclusively on sensing nodes and server) and mobile devices are monitored passively, the user has no control over being located beyond turning off their WiFi or stopping to communicate with our server. On the other hand, our technique does not employ continuous tracking and does not rely on storing historical data beyond a couple of seconds to compute trailing averages. Furthermore, monitoring the wireless traffic has the aim to explicitly provide a service to the user based on the proximity detection.

CONCLUSION AND FUTURE WORK

In this paper we have presented a technique based on *proximity-adaptive HTTP responses* to bring WiFi proximity detection to web applications without having to install non-standard software on the client device. Through ProxiMagic we have demonstrated that PAHR can be realized with low-cost hardware and providing adequate proximity detection even with a simplistic proximity detection scheme. Local Area Artworks demonstrates the real-world applicability of the technique, and motivates the need for zero-install proximity detection.

There are numerous potential application domains that could make use of anchoring a digital layer to a physical space at certain points of interest. This includes located information, discussions or advertisements. But it also includes access to physical resources, whether to control the servo of nearby window blinds or to diagnose a piece of equipment on the factory floor. In the future, we seek to integrate the technique into i) a whiteboard capture system that allows mobile access to captured content of different whiteboards by being in the respective room, into ii) a system for the public library to provide section-specific information but also localized discussions and literature recommendations, and into iii) multi-surface environments where the technique could allow the pairing of mobile devices to stationary surfaces in order to act as remote input devices or to transfer content between devices.

ACKNOWLEDGMENTS

This work is funded by the Center for Participatory IT at Aarhus University, and by the Danish Council for Strategic Research through the EcoSense project. We thank Janus Bager Kristensen and Rolf Bagge from the Center for Advanced Visualization and Interaction (CAVI) at Aarhus University for being instrumental in the development and testing of ProxiMagic. We thank Anna Maria Polli for observations and interviews, and Kunsthal Aarhus for their cooperation in the project.

REFERENCES

1. Aalto, L., Göthlin, N., Korhonen, J., and Ojala, T. Bluetooth and WAP push based location-aware mobile advertising system. In *Proc. MobiSys 2004*, ACM Press (2004), 49–58.
2. Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless Networks 3*, 5 (1997), 421–433.
3. Bahl, P., and Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In *Proc. INFOCOM 2000*, vol. 2, IEEE (2000), 775–784.
4. Benford, S., Crabtree, A., Flintham, M., Drozd, A., Anastasi, R., Paxton, M., Tandavanitj, N., Adams, M., and Row-Farr, J. Can you see me now? *ACM Transactions on Computer-Human Interaction 13*, 1 (2006), 100–133.
5. Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proc. CHI 2000*, ACM Press (2000), 17–24.
6. Kjærgaard, M. B., Blunck, H., Godsk, T., Toftkjær, T., Christensen, D., and Grønbæk, K. Indoor positioning using GPS revisited. In *Proc. Pervasive 2010*, Springer (2010), 38–56.
7. Kjærgaard, M. B., Krarup, M. V., Stisen, A., Prentow, T. S., Blunck, H., Grønbæk, K., and Jensen, C. S. Indoor positioning using wi-fi – how well is the problem understood? In *Proc. Indoor Positioning and Indoor Navigation (IPIN 2013)*, IEEE (2013).
8. Kjærgaard, M. B., Treu, G., Ruppel, P., and Küpper, A. Efficient indoor proximity and separation detection for location fingerprinting. In *Proc. MOBILWARE 2008*, ICST (2007), 1:1–1:8.
9. Krumm, J., and Hinckley, K. The NearMe wireless proximity server. In *Proc. UbiComp 2004*, Springer (2004), 283–300.
10. Liu, H., Darabi, H., Banerjee, P., and Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37*, 6 (2007), 1067–1080.
11. Polli, A. M., Korn, M., and Klokmose, C. N. Local Area Artworks: Collaborative art interpretation on-site. In *Adjunct Proc. UbiComp 2013*, ACM Press (2013), 79–82.
12. Popescu, A. Geolocation api specification. *World Wide Web Consortium, Candidate Recommendation CR-geolocation-API-20100907* (2010).
13. Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., and Otsason, V. GSM indoor localization. *Pervasive and Mobile Computing 3*, 6 (2007), 698–720.
14. Want, R., Hopper, A., Falcão, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems 10*, 1 (1992), 91–102.
15. Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., and Weiser, M. An overview of the PARCTAB ubiquitous computing experiment. *IEEE Personal Communications 2*, 6 (1995), 28–43.
16. Ward, A., Jones, A., and Hopper, A. A new location technique for the active office. *IEEE Personal Communications 4*, 5 (1997), 42–47.