Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances

Robert Xiao **Gierad Laput** **Yang Zhang** **Chris Harrison**
Carnegie Mellon University, Human-Computer Interaction Institute
5000 Forbes Ave, Pittsburgh, PA 15213
{brx, gierad.laput, yang.zhang, chris.harrison}@cs.cmu.edu

ABSTRACT
Homes, offices and many other environments will be increasingly saturated with connected, computational appliances, forming the "Internet of Things" (IoT). At present, most of these devices rely on mechanical inputs, webpages, or smartphone apps for control. However, as IoT devices proliferate, these existing interaction methods will become increasingly cumbersome. Will future smart-home owners have to scroll though pages of apps to select and dim their lights? We propose an approach where users simply tap a smartphone to an appliance to discover and rapidly utilize contextual functionality. To achieve this, our prototype smartphone recognizes physical contact with uninstrumented appliances, and summons appliance-specific interfaces. Our user study suggests high accuracy – 98.8% recognition accuracy among 17 appliances. Finally, to underscore the immediate feasibility and utility of our system, we built twelve example applications, including six fully functional end-to-end demonstrations.

Author Keywords
Object Sensing; Internet of Things; IoT; EMI; Context Sensing; Smart Appliances; Smart Home; Recognition.

ACM Classification Keywords
H.5.2. User Interfaces: Input devices and strategies.

INTRODUCTION
We are surrounded by an ever-growing ecosystem of connected and computationally-enhanced appliances, from smart thermostats and light bulbs, to coffee makers and refrigerators. The much-lauded "Internet of Things" (IoT) revolution predicts billions of such devices in use by the close of the decade [8]. Despite offering sophisticated functionality, most IoT devices provide only rudimentary on-device controls. This is because 1) it is expensive to include *e.g.*, large touchscreen displays on low-cost, mass-market hardware, and 2) it is challenging to provide a full-featured user experience in a small form factor. Instead, most IoT appliances rely on users to launch special-purpose applications on their smartphones or browse to a specific webpage in the cloud or on their local area network. Quintessential examples include "smart" light bulbs (*e.g.*, Philips Hue), media devices (*e.g.*, Chromecast), Wi-Fi cameras (*e.g.*, Dropcam) and internet routers.

Clearly, this manual launching approach will not scale as the number of IoT devices grows. If we are to have scores of these devices in our future homes and offices—as many prognosticate—will we have to search through scores of applications to dim the lights in our living room or find something to watch on TV? What is needed is an instant and effortless way to automatically summon rich user interface controls, as well as expose appliance-specific functionality within existing smartphone applications in a contextually relevant manner.

In this paper, we explore two approaches to mitigate the aforementioned interaction bottleneck. The most straightforward option is to automatically launch manufacturers' applications *instantly* upon contact with the associated appliance. For example, touching a smartphone to a thermostat would launch the thermostat's configuration app (Figure 4). In this case, the currently running app on the phone is swapped out for a new full screen app. Alternatively, the phone can expose what we call *contextual charms*—small widgets that allow the currently running smartphone application to perform actions *relevant to the touched appliance*. For example, if reading a PDF, touching the phone to a printer will reveal an on-screen print button (Figure 1).

This general vision of rapid and seamless interaction with connected appliances has been explored many times in the literature (*e.g.*, [16, 23, 29]), and in this work we set out to practically achieve it. We created full-stack implementations for several example applications to show that the interactions are realizable today. In cases where appliances have proprietary APIs, we can automatically launch the manufacturers' app. For IoT devices with open APIs (fortunately the trend), contextual charms can expose appliance-specific functionality across the smartphone experience.

To recognize appliances on-touch, we had to significantly extend the technical approach used in our prior work, EM-Sense [17] – a smartwatch that detected electromagnetic emissions of grasped electrical and electromechanical ob-

ACM 978-1-4503-4655-9/17/05 $15.00
DOI: http://dx.doi.org/10.1145/3025453.3025828

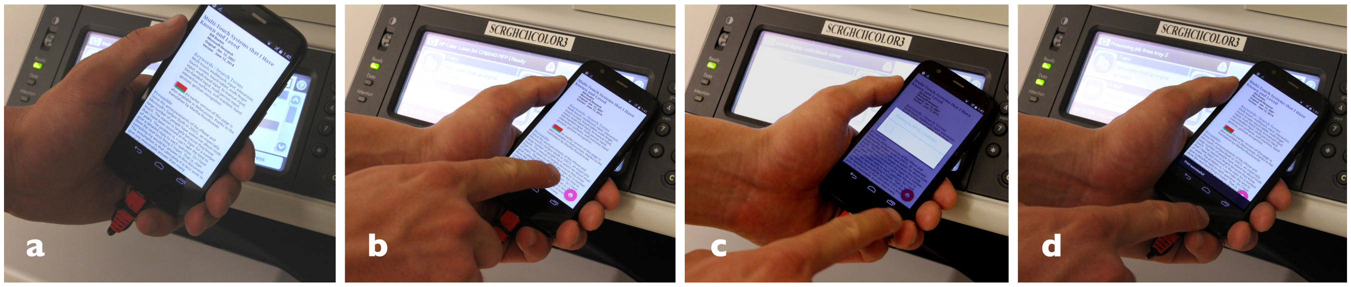

Figure 1. When a user is reading a document (a), they can tap the phone to a printer to bring up a "print" contextual charm (b). Pressing the charm spools the document to the printer (c), which prints it (d).

jects. As appliances are typically long-lived and rarely replaced, the adoption rate of "smart appliances" is expected to be slow – few, if any appliances in the average home today are *e.g.* NFC-enabled. Thus, our technical approach requires no modification or instrumentation of appliances, and can therefore work "out of the box" with already-deployed devices.

In summary, we contribute a novel system that allows instant recognition of uninstrumented electronic appliances, which in turn allows us to expose contextual functionality via a simple tap-to-device interaction. We demonstrate high accuracy (98.8% recognition of 17 unmodified appliances in our user study), while running entirely on an augmented smartphone. In addition to conventional full-screen applications, we contribute contextual charms, a new cross-device interaction technique. Finally, in contrast to most prior work, we created truly functional implementations for many of our example demos using public IoT APIs.

RELATED WORK

We now briefly review two key categories of related work. First are systems that tackle the stubborn problem of inter-device control, especially those that use a mobile device as the portal. Secondly, we describe research relating to our technical approach.

Recognizing and Controlling Appliances

Using mobile devices to control appliances has been explored many times in prior work. An early system by Hodes *et al.* [16] allowed users to control multiple pieces of lecture hall equipment from a single wireless laptop, though users still had to manually select the target device from a graphical map. To alleviate this manual selection process, later work has considered a bevy of technical approaches to automatically select and recognize appliances from mobile devices, including RFID tags [27, 28, 31], fiducial tags [15], near-field communication (NFC) [9], laser pointers [2, 26], handheld projectors [29] and personal area networks [25]. These systems allow users to select appliances by tapping or pointing, but of course require appliances (or the environment) to be specially instrumented with tags or sensors working in concert with custom emitters or sensors.

For example, Reality Editor [15] is an augmented-reality smartphone app that allows users to rapidly link functionality and/or controls across devices (*e.g.*, linking music volume to a knob). Objects are recognized via fiducial markers applied to their enclosures, and then device-specific functionality is overlaid onto the object. Another compelling example is PICOntrol [29], which allows users to point a handheld projector at an object, providing both a graphical interface and command transmission through structured light. However, the object must be instrumented with a receiver photodiode and processor to decode commands. Besides these device-to-appliance systems, there is an entire body of related work on device-to-device pairing [4].

Although many of the above systems facilitate recognition of appliances, few demonstrate *actual* control of real appliances (*e.g.*, using IoT APIs). Closer to our work are "full-stack" systems (*i.e.*, recognition up through appliance control) that do not require instrumentation of the appliance. GeeAir [24] allows users to select appliances by either pointing at them with a handheld control or by speaking the name of the appliance. Proximic-Aware Controls [18] used the orientation and position of a user's phone to demonstrate appliance interactions that were sensitive to physical context (*e.g.,* distance to appliance) but not digital context (*e.g.,* user's active application or document). Mayer *et al.* [22] demonstrated a system that uses a smartphone's camera (combined with machine learning) to classify objects in the environment and overlay a suitable control interface. This system demonstrated classification between eight different objects, although no formal accuracy evaluation was presented. Finally, Snap-To-It [7] also used a smartphone camera in conjunction with a database of appliance images to automatically classify appliances and summon appropriate interfaces. The result is much like what we propose, although the technical approach is very different.

Electromagnetic Emissions Sensing

Most closely related to our technical approach are methods that perform recognition by sensing the electromagnetic emissions of electrical, electromechanical and electronic devices. Such radiation (and EM noise in general) is most often a byproduct of normal operation. Because of this, it is possible to sense appliance operation by sensing "electrical events" on power lines [1, 11, 13, 14].

It is also possible to use the "body as an antenna", as proposed in LightWave [10] and in Cohn *et al.* [6]. The latter system was able to recognize touch events on unin-

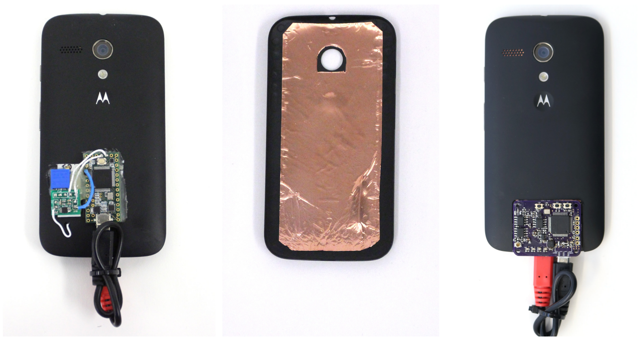
Figure 2. Our sensing hardware attached to phone. Left: original prototype. Center: copper tape antenna inlaid on rear cover. Right: final prototype with integrated PCB.

strumented walls and free-space gestures based on body coupling to noisy power lines. Superior gesture recognition accuracy was achieved in the follow up system, Humantenna [5]. uTouch [3] used EM coupling to detect touch gestures on LCD displays. Similarly, EMI-Spy [34] used EM signatures to distinguish different LCD displays, enabling pointing and touching gestural input. These projects all used external laptop and data acquisition units, with sampling rates ranging from 100 kHz to 1 MHz.

Closer to the present work are methods that attempt to recognize handheld objects. For example, Maekawa *et al.* explored using magnetometers [20] and hand-worn coils [21] to detect grasped objects based on changes in the oscillating magnetic field. Wang *et al.* propose a similar magneto-inductive system called MagnifiSense [30]. Lastly, EM-Sense [17] used the body antenna effect to recognize electrical devices on-touch or on-activation. The prototype is worn as a smartwatch, though it uses a software defined radio (SDR) connected to a smartphone over USB for sensing, which then streams data to a laptop for classification. This prototype sampled at 1 MHz with an 8-bit resolution.

In contrast to these systems, our sensing hardware is compact, inexpensive, runs on a low-power embedded processor, and offers much higher sampling rate and resolution: 4.36 MHz at 12-bit resolution. Furthermore, all processing and recognition takes place on the phone itself, making integration into consumer devices significantly more feasible.

IMPLEMENTATION

Our system consists of a smartphone instrumented with a compact EM sensor (Figure 2). All signal processing and classification is performed in real-time on the phone. Further, all components are powered from the smartphone's battery, making it a fully self-contained system.

Prototype Smartphone

Our smartphone prototype consists of an instrumented Moto G XT1031, a mid-tier 2015 Android phone costing around $100 USD (Figure 2). This phone has a 1.2 GHz quad-core Snapdragon processor and 1 GB of RAM. It features a removable plastic rear cover, which we inlaid with copper tape to serve as our antenna (Figure 2, center). The antenna was made as large as possible to maximize signal coupling and pick up low-frequency EM (0-2 MHz). In a commercial version, it may be possible to utilize an existing antenna or utilize the phone's internal chassis (which is often made of magnesium or aluminum).

EM Sensing

The antenna is connected to a 50x inverting amplifier circuit compactly mounted on a custom PCB. This circuit amplifies the weak EM signals and adds a 1.6 V DC bias to move the signal to the 0-3.3 V range of our analog-to-digital converter (ADC). The amplified signal is then sampled by an ARM Cortex-M4 microcontroller running at 96 MHz (MK20DX256VLH7).

We sample the analog signal with 12-bit resolution at 4.36 MHz. We achieve this high sampling rate by running both of the microcontroller's ADCs on the same pin with interleaved triggers. We use the microcontroller's direct memory access (DMA) unit to copy the ADC samples to main memory, reducing processor overhead. The total cost of the integrated sensing hardware shown in Figure 2, right (*i.e.*, PCB, antenna, OpAmp, microcontroller and misc. passives) is under $10 in volume.

Embedded Processing

The first stage of data processing takes place on the microcontroller itself. The processor continuously runs 1024-sample discrete Fourier transforms (DFTs) on the input signal to extract the frequency spectra. We take the magnitude of the resulting complex-valued spectra to obtain amplitude spectra. Using an optimized, 16-bit fixed-point real-valued DFT, the processor performs ~1000 transforms per second.

To improve the stability of the frequency domain data, we track the frequency-wise maximum over a running 40 ms window. We use a running maximum, rather than an average, in order to capture the transient signals typical of digital devices. Finally, the maximum amplitude spectra are reported to the phone at 100 FPS using an On-The-Go (OTG) USB connection. This link is used both for transmitting data and for providing power to our sensing board.

Recognition

Our object recognition pipeline runs on the Android phone as a background service. The basic implementation largely follows the approach described in EM-Sense [17].

For each spectrum captured by the embedded processor, we extract a set of 699 features: the 512-element amplitude spectrum, the indices of the minimum and maximum spectrum elements, the root-mean-square (RMS) measurement, the mean and standard deviation of the spectrum, and pairwise band ratios. In particular, due to limited computational resources on the phone, we do not compute features over the 1st or 2nd derivatives, nor the 2nd-order FFT.

Next, the features are fed to an ensemble of 153 binary linear-kernel support vector machine (SVM) classifiers, one for each possible pairing of the 18 output classes. The ensemble's output is determined through plurality voting. The entire classification process, including feature calculation, takes about 45 ms. We use the Weka machine learning toolkit [12], which we modified to run on Android, to perform classification on the phone.

Finally, the classification is stabilized by outputting the most common classification amongst a window of the last 20 ensemble outputs. This voting scheme ensures that spurious or intermittent electrical signals do not result in errant classifications. In particular, without voting, "intermediate" signatures produced while a phone moves towards an object could result in incorrect classifications. This voting scheme introduces around 450 ms of latency into the pipeline. Once an object is recognized, the service displays a contextual charm on screen (if the foreground application supports it) or launches the appropriate full-screen control app.

Contextual Charms

Our contextual charms are small floating buttons that appear along the right edge of an application when the phone touches a supported device. These relate to both the context of the running smartphone application *and* the touched appliance. These buttons trigger specific phone-to-device actions, expressed as verb-object pairs: "print document", "copy text", "scan document", "cast audio", and so on. The charm application framework predefines several verbs and object types, and leaves room for future expansion.

Our charm service runs as a background Android service alongside the classifier. Appliance and smart device drivers advertise the set of supported actions to the charm service (*e.g.* the HP printer driver registers the "print document" action on all supported printer models), which maintains a central registry. User-facing applications report the actions they can currently perform to the charm service.

When an object's EM signature is detected, the charm service matches the object's supported actions to available application actions, then informs the application that new contextual actions are available. Within the application, selecting an action dispatches an "execute" command to the service, which in turn dispatches the verb and associated object data to the object's appliance driver (*e.g.,* a MediaRouter instance to implement casting of an audio file, or a backend printing driver to handle a document file). In this way, the charm service abstracts physical objects into receivers for application actions, allowing application developers to easily target arbitrary devices without needing to know specific device details.

We envision that future smart appliance applications would register their device's EM signature and a set of verbs with the charm system service upon installation, which would enable existing apps to immediately take advantage of appliances and devices in a user's environment. This is analogous to the current paradigm of applications registering Android "share" handlers to support system-wide sharing of content to *e.g.,* social media.

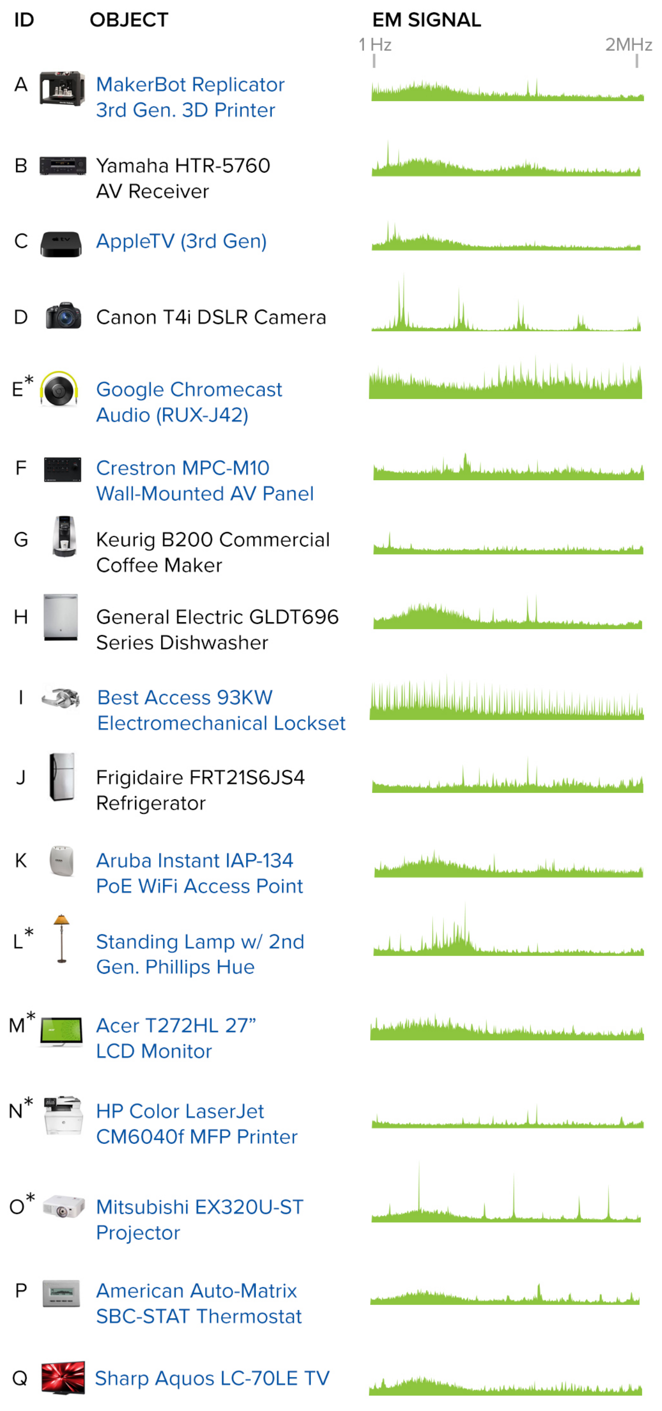

Figure 3. The appliances we identified in our lab that typified poor access to functionality, along with their "EM signatures". Appliance names in blue signify the device offers some type of connectivity for external control. Appliances in grey offered no current connectivity, but it is easy to imagine future incarnations that do. We created demos for eleven appliances to illustrate the interactions our technique enables. For appliances with open or known APIs, we created fully functional demos (asterisks).

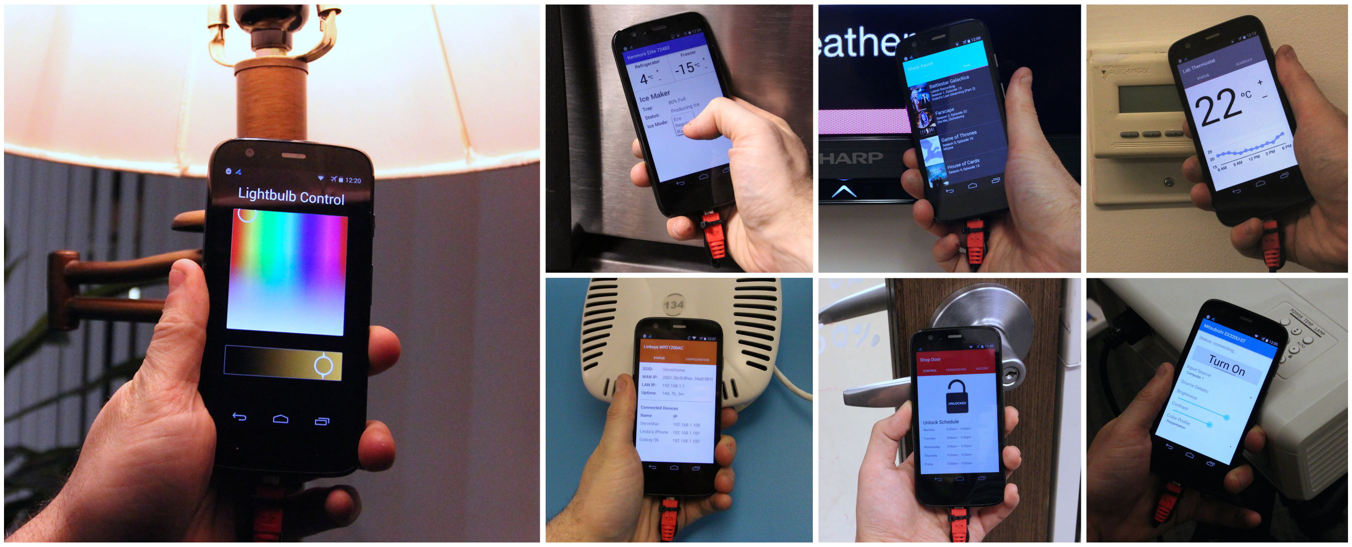

Figure 4. Example full-screen applications. Clockwise from left: lightbulb, refrigerator, television, thermostat, projector, door lock, and wireless router. Please also see Video Figure.

Duplicate Appliance Disambiguation

As discussed in EM-Sense [17], all devices of the *exact* same model produce the similar EM signatures (though not identical, as discussed in [33]). Generally, this is not a substantial issue because people own a single instance of most appliances. For example, it is unlikely for a single household to have multiple refrigerators, let alone multiple of the same model. However, a single home (or office building, etc.) could have many thermostats or smart light bulbs of the same model. Although there are small EM differences that are characteristic of particular locations, we found these to be too unreliable for robust disambiguation. One possible solution is to rely on smartphones' geo-location reporting to disambiguate which of several appliances the user is proximate to (if there are multiple in a single context). Although indoor geo-location is currently coarse, much work is being done to improve accuracy [19].

EXAMPLE APPLIANCES

We identified 17 appliances in our lab that typified poor access to rich functionality (Figure 3). Eleven of these devices (names in blue) have some form of connectivity, though their "smartness" varies. For example, the Apple TV is connected to WiFi and can be controlled through a cloud portal or via iOS devices. On the other hand, our building's HVAC system is wired and computer controlled, but not accessible for external control in the traditional IoT sense.

We also include five appliances with no connectivity (Figure 3, names in grey), which serve as stand-ins for future "smart" versions of themselves. For example, we include a Keurig B200, a basic coffee brewing machine with no IoT functionally, as a proxy for future smart coffee makers. Although this lack of connectivity prevents us from building fully functional control implementations, it nonetheless allows us to explore how interactions with these devices might feel if there were to be made smart in the future.

EXAMPLE FULL-SCREEN APPS

We built three apps to illustrate controlling common infrastructure hardware, and four apps to demonstrate control of common appliances. Please see Figure 4 and Video Figure for illustrations of all full screen applications.

Infrastructure Hardware

One of the most painful interactions on contemporary thermostats is setting a heating/cooling schedule. In response, we built a multi-pane configuration application for our building's thermostats, which instantly launches when a phone is tapped to any unit. Another awkward interaction is router control, which requires remembering and typing a numerical IP address (*e.g.* "192.168.1.1") to access the web-based configuration panel. To simplify this process, we launch a router control application when the phone touches the wireless router; the IP address to the router could be inferred automatically from the gateway address on the device's WiFi connection. Finally, some areas of our building require ID card access. Reviewing access control and history is tedious; we made an app that would offer easy access to both.

Although these apps are meant to be illustrative, they are unfortunately non-functional as we currently do not have access to the control systems used to manage these devices.

Household Appliances

For our Frigidaire refrigerator, our app displays the temperatures set for the main and freezer compartments, as well as the status and mode of the icemaker. For our television set, we built a "remote control" app that allows users to control the TV's input source and manage the built-in DVR functionality. These two devices have no network connectivity at all (although networked smart fridges and TVs do exist), so the applications are simply illustrative.

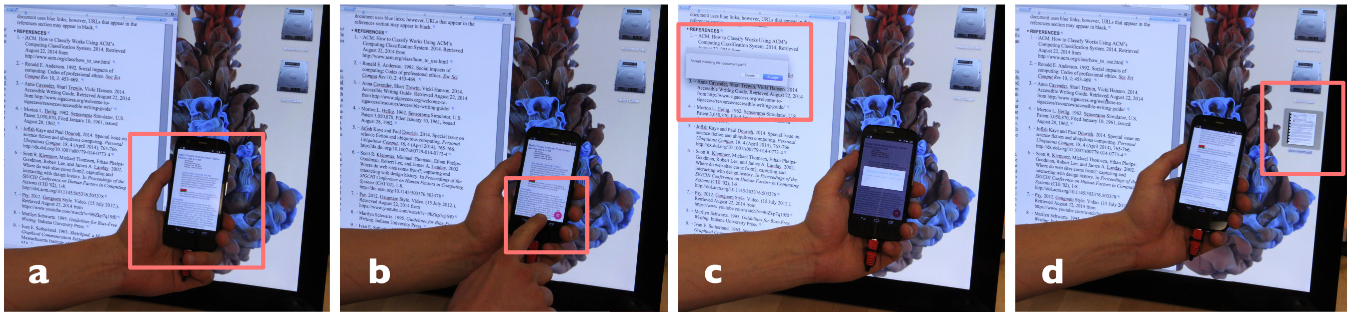

Figure 5. "Send Document" contextual charm. With a document open (a), the user taps the phone to the screen to get a "send" charm (b). Activating the charm transmits the document to the computer (c, d).

Finally, we also built two fully functional applications on top of device APIs. For the Philips Hue light bulb, users can touch the phone to any part of the metal standing lamp to trigger the full screen control app. The app connects to the Philips Hue wireless bridge device through UPnP autodiscovery, and then issues commands using the Hue's REST API to control the color and brightness of the lightbulb in response to user input.

For the Mitsubishi projector, users can touch the phone to the side of the device to launch a fully functional configuration app. The app connects to the projector over the Crestron device control protocol to obtain the device status and issue control commands. Our app can change the input source, power the projector on and off, and adjust the brightness and contrast of the image.

EXAMPLE CONTEXTUAL CHARMS

We also created five contextual charms to demonstrate targeted interactions with devices tailored for specific applications. Four of these charms are fully functional, illustrating the immediate feasibility of this technique.

In our prototype document reader, users can tap their phone to a computer monitor, bringing up a "send" charm (Figure 5). Pressing the charm uploads the current document to the computer's desktop. The charm establishes a connection with an authenticated custom file transfer service on the local network, with the user confirming the file transfer on the computer. The file transfer dialog could appear on any computer the user owns, and this action of confirming the transfer ensures the file is sent to the right device.

Users can also select a segment of text on the phone to reveal a "copy" charm when pressed to a computer monitor (Figure 6). If activated, the text is copied to the computer's clipboard. This uses the same file transfer service, but instead copies the sent data to the clipboard using the Mac OS X *pbcopy* utility. This can be used not only for text, but also media, such as PDFs and images.

Tapping the phone to a TV reveals a "cast" charm (Figure 7), which can be used to show the current document on the larger screen. This charm was only illustrative, as we did not have a suitable networked TV for automatic casting.

When the phone is tapped to a printer, it brings up a "print" charm (Figure 1), which spools the current document to the printer. The document is rendered using the Android printing API, and sent to our institution's networked print spooler over WiFi. The printer can be identified through its EM signature (unique per model) and geo-location.

Finally, we also built a music player app. When the phone is touched to a Chromecast Audio device, a "cast" charm appears on-screen (Figure 8). Tapping this charm automatically transfers the audio stream to the Chromecast. This is implemented using the Android Media-Router remote streaming API, and assumes that only the phone and Chromecast Audio have been previously paired.

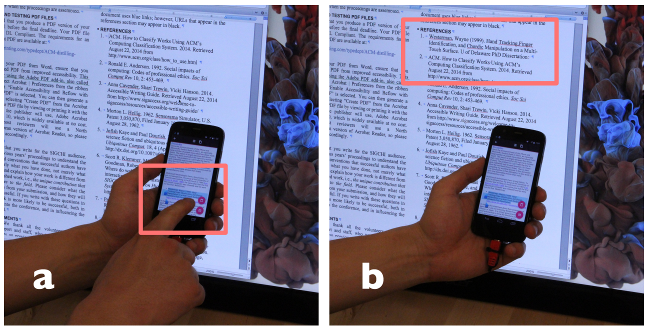

Figure 6. "Copy Text" contextual charm. With text selected, a second "copy" charm appears (a). Tapping that charm copies the text to the tapped computer's clipboard, which the user then pastes into the document (b).

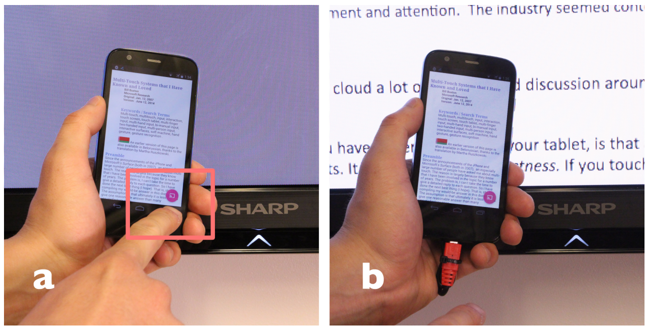

Figure 7. "Cast Video" contextual charm. With a document open, the user touches the phone to a TV to reveal a "cast" charm (a). Tapping this charm casts this document in full resolution to the TV (b).

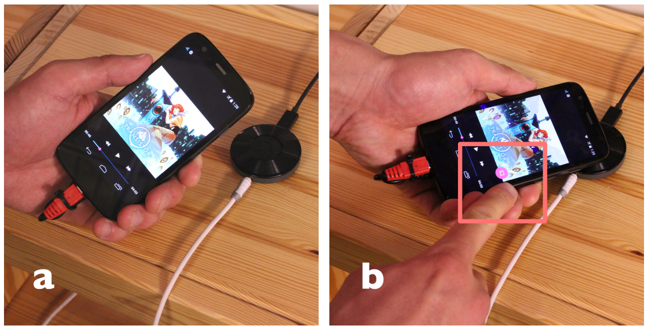

Figure 8. "Cast Audio" contextual charm. While the user is playing an audio file (a), they can tap the phone to a Chromecast Audio device to see a "cast" charm (b). Tapping on that charm transfers the current audio stream to the Cast device' connected to the room's speakers.

EVALUATION

Prior to the study, we trained our smartphone prototype (Figure 2, left) to recognize appliances. Specifically, we held the smartphone to a device's surface and collected 500 EM signature instances over a five second period. This procedure was repeated for each of our 17 example appliances in a random order. A second round of data was collected on a different day (to mitigate potential over-fitting to environmental conditions). We also collected six rounds (3000 instances) of "no appliance", *i.e.*, ambient background EM noise. We then used this data (20,000 instances in total) to train and deploy an object-recognition classifier to the smartphone.

We recruited 10 participants (five female, mean age 28.6, mean BMI=23.2) for our user study, which took approximately 30 minutes to complete and paid $10. Our 17 example appliances were located in five zones: common area, conference room, kitchen, office, and living room. Participants visited these zones in a random order. Within each zone, users touched the smartphone to one appliance at a time. Each appliance was requested three times, and the order of requests was randomized. In total, this yielded 510 trials (10 participants x 17 appliances x 3 repeats).

Of note, the smartphone performed *live*, on-device classification (*i.e.*, no post hoc feature engineering, kernel parameter optimization, etc.). Furthermore, there was no per-user calibration or training – a single, *pre-trained* classifier was used throughout the experiment and across all participants. Although still a lab study, this practice more closely emulates real world deployment (where a classifier might be deployed to many devices with an over-the-air update). In addition to using a classifier trained more than a week prior, we also ran our user study over a three-day period, demonstrating the temporal stability of our system.

RESULTS

Across 10 users and 17 objects, our system achieved an overall accuracy of 98.8% (SD=1.7%). 14 objects achieved 100% accuracy. The three non-perfect devices were our Wall-Mounted AV panel (Object F) and Thermostat (P), which were 96% accurate, and our Standing Lamp with Phillips Hue Bulb (L) at 86% accuracy. We suspect the lamp recognition errors were due to us allowing participants to touch anywhere on the metal lamp stand, as opposed to the bulb itself. Nonetheless, our system was fairly robust overall, and we found no relationship on system accuracy across users, location or time.

LIMITATIONS

We initially set out to produce full-stack implementations for all of the network-connected devices on our list. However, we were stymied by the lack of public APIs on several of them. Furthermore, even when APIs were available, some were vendor-locked (*e.g.,* the Apple TV casting APIs were only open to Apple devices). In order for the future Internet of Things to have true impact, open APIs are a strong requirement. Until then, our system will be limited by the inability to talk to all smart devices.

It is also important to note that our approach cannot detect objects when they are truly and totally powered off. However, many objects have a detectable "low-power" or "sleep" mode when not active (see e.g., dishwasher EM signal in Video Figure), particularly if they have an always-on Internet connection. More generally, for robust detection in all circumstances, our system would have to be trained with a representative sample of possible device states.

In large, shared environments, our system cannot accurately distinguish between multiple instances of a particular object (*e.g.* multiple thermostats). Fundamentally, distinguishing objects that are outwardly identical is very difficult, and in this work we make our best effort via techniques such as network broadcasting (*e.g.* UPnP, broadcast-and-confirm) and geolocation. However, there is some promise in using minute differences in devices' EM signatures to distinguish even apparently identical devices [33], which is an area for future exploration.

Finally, as with any EM sensing approach, our system is susceptible to external interference from powerline noise. Such noise can confuse the classifier and decrease accuracy. This is a hard problem to solve, and was previously acknowledged in EM-Sense. Future designs should implement strong noise rejection to reject common sources of powerline noise (*e.g.* high-current devices such as vacuum cleaners), but further work remains to be done.

CONCLUSION

We have presented a novel system that enables users to simply tap their smartphone to an appliance to interact with it. To achieve this, we developed a novel hardware sensing configuration and combined it with efficient and accurate real-time classification to create a self-contained prototype. Through a user study, we quantified the characteristics of our system. Finally, we demonstrated a number of applications enabled by our technique, including several full-stack implementations, thus providing a preview of what our future "smart" world might feel like in the near future.

ACKNOWLEDGMENTS
This research was generously supported by the David and Lucile Packard Foundation, a Google Faculty Research Award, and Qualcomm.

REFERENCES
1. R.E. Abbott and S.C. Hadden, Product Specification for a Nonintrusive Appliance Load Monitoring System. EPRI Report #NI-101, 1990.
2. Michael Beigl. 1999. Point & click-interaction in smart environments. In *Handheld and Ubiquitous Computing*, 311–313. http://dx.doi.org/10.1007/3-540-48157-5_31
3. Ke-Yu Chen, Gabe A. Cohn, Sidhant Gupta, and Shwetak N. Patel. 2013. uTouch: sensing touch gestures on unmodified LCDs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 2581-2584. http://dx.doi.org/10.1145/2470654.2481356
4. Ming Ki Chong, Rene Mayrhofer, and Hans Gellersen. 2014. A Survey of User Interaction for Spontaneous Device Association. *ACM Comput. Surv.* 47, 1, Article 8 (May 2014), 40 pages. http://dx.doi.org/10.1145/2597768
5. Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 1901-1910. http://dx.doi.org/10.1145/2207676.2208330
6. Gabe Cohn, Daniel Morris, Shwetak N. Patel, and Desney S. Tan. 2011. Your noise is my command: sensing gestures using the body as an antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 791-800. http://dx.doi.org/10.1145/1978942.1979058
7. Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '16), 5909-5920. http://dx.doi.org/10.1145/2858036.2858177
8. Gartner, Inc. 2015. "Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016." November 10, 2015. http://www.gartner.com/newsroom/id/3165317
9. Arjan Geven, Peter Strassl, Bernhard Ferro, Manfred Tscheligi, and Harald Schwab. 2007. Experiencing real-world interaction: results from a NFC user experience field trial. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services* (MobileHCI '07). ACM, New York, NY, USA, 234-237. http://dx.doi.org/10.1145/1377999.1378012
10. Sidhant Gupta, Ke-Yu Chen, Matthew S. Reynolds, and Shwetak N. Patel. 2011. LightWave: using compact fluorescent lights as sensors. In *Proceedings of the 13th international conference on Ubiquitous computing* (UbiComp '11). ACM, New York, NY, USA, 65-74. http://dx.doi.org/10.1145/2030112.2030122
11. Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the 12th ACM international conference on Ubiquitous computing* (UbiComp '10). ACM, New York, NY, USA, 139-148. http://dx.doi.org/10.1145/1864349.1864375
12. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
13. George W. Hart. Advances in Nonintrusive Appliance Load Monitoring. In *Proc. EPRI Information and Automation Conference* '91.
14. George W. Hart. Nonintrusive appliance load monitoring. In *Proc. of the IEEE*, 1992. 1870-1891.
15. Valentin Heun, James Hobin, and Pattie Maes. 2013. Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (UbiComp '13 Adjunct). ACM, New York, NY, USA, 307-310. http://dx.doi.org/10.1145/2494091.2494185
16. Todd D. Hodes, Randy H. Katz, Edouard Servan-Schreiber, and Lawrence Rowe. 1997. Composable ad-hoc mobile services for universal interaction. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking* (MobiCom '97). ACM, New York, NY, USA, 1-12. http://dx.doi.org/10.1145/262116.262121
17. Gierad Laput, Chouchang Yang, Robert Xiao, Alanson Sample, and Chris Harrison. 2015. EM-Sense: Touch Recognition of Uninstrumented, Electrical and Electromechanical Objects. In*Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). ACM, New York, NY, USA, 157-166. http://dx.doi.org/10.1145/2807442.2807481
18. David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. 2015. Proxemic-Aware Controls: Designing Remote Controls for Ubiquitous Computing Ecologies. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15). ACM, New York, NY, USA, 187-198. http://dx.doi.org/10.1145/2785830.2785871

19. Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. 2015. A realistic evaluation and comparison of indoor location technologies: experiences and lessons learned. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks* (IPSN '15). ACM, New York, NY, USA, 178-189. http://dx.doi.org/10.1145/2737095.2737726
20. Takuya Maekawa, Yasue Kishino, Yasushi Sakurai, and Takayuki Suyama. 2011. Recognizing the use of portable electrical devices with hand-worn magnetic sensors. In *Proceedings of the 9th international conference on Pervasive computing* (Pervasive'11), Kent Lyons, Jeffrey Hightower, and Elaine M. Huang (Eds.). Springer-Verlag, Berlin, Heidelberg, 276-293.
21. Takuya Maekawa, Yasue Kishino, Yutaka Yanagisawa, and Yasushi Sakurai. 2012. Recognizing handheld electrical device usage with hand-worn coil of wire. In *Proceedings of the 10th international conference on Pervasive Computing* (Pervasive'12), Judy Kay, Paul Lukowicz, Hideyuki Tokuda, Patrick Olivier, and Antonio Krüger (Eds.). Springer-Verlag, Berlin, Heidelberg, 234-252. http://dx.doi.org/10.1007/978-3-642-31205-2_15
22. Simon Mayer, Markus Schalch, Marian George, and Gábor Sörös. 2013. Device recognition for intuitive interaction with the web of things. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (UbiComp '13 Adjunct). ACM, New York, NY, USA, 239-242. http://dx.doi.org/10.1145/2494091.2494168
23. Dan Olsen. 2008. Interactive viscosity. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST '08). ACM, New York, NY, USA, 1-2. http://dx.doi.org/10.1145/1449715.1449717
24. Gang Pan, Jiahui Wu, Daqing Zhang, Zhaohui Wu, Yingchun Yang, and Shijian Li. 2010. GeeAir: a universal multimodal remote control device for home appliances. *Personal Ubiquitous Comput.* 14, 8 (December 2010), 723-735. http://dx.doi.org/10.1007/s00779-010-0287-7
25. Duck Gun Park, Jin Kyung Kim, Jin Bong Sung, Jung Hwan Hwang, Chang Hee Hyung, and Sung Weon Kang. 2006. TAP: touch-and-play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), Rebecca Grinter, Thomas Rodden, Paul Aoki, Ed Cutrell, Robin Jeffries, and Gary Olson (Eds.). ACM, New York, NY, USA, 677-680. http://dx.doi.org/10.1145/1124772.1124873
26. Shwetak N. Patel and Gregory D. Abowd. 2003. A 2-way laser-assisted selection scheme for handhelds in a physical environment. In *UbiComp '03*, 200–207. http://dx.doi.org/10.1007/978-3-540-39653-6_16
27. Enrico Rukzio, Karin Leichtenstern, Vic Callaghan, Paul Holleis, Albrecht Schmidt and Jeannette Chin. 2006. An experimental comparison of physical mobile interaction techniques: touching, pointing and scanning. In Proceedings of the 8th international conference on Ubiquitous Computing (UbiComp'06), Paul Dourish and Adrian Friday (Eds.). Springer-Verlag, Berlin, Heidelberg, 87-104. http://dx.doi.org/10.1007/11853565_6
28. Ichiro Satoh. 2011. A Management Framework for Context-Aware Multimedia Services.. In *DMS*. 165–170.
29. Dominik Schmidt, David Molyneaux, and Xiang Cao. 2012. PICOntrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, 379-388. http://dx.doi.org/10.1145/2380116.2380166
30. Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. 2015. MagnifiSense: inferring device interaction using wrist-worn passive magneto-inductive sensors. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '15). ACM, New York, NY, USA, 15-26. http://dx.doi.org/10.1145/2750858.2804271
31. Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. 1999. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99). ACM, New York, NY, USA, 370-377. http://dx.doi.org/10.1145/302979.303111
32. Jong-bum Woo and Youn-kyung Lim. 2009. Contact-and-connect: designing new pairing interface for short distance wireless devices. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '09). ACM, New York, NY, USA, 3655-3660. http://doi.acm.org/10.1145/1520340.1520550
33. Chouchang Yang and Alanson P. Sample. 2016. EM-ID: Tag-less identification of electrical devices via electromagnetic emissions. In *Proceedings of the 2016 International Conference on RFID* (RFID '16). IEEE, 8 pages. http://dx.doi.org/10.1109/RFID.2016.7488014
34. Nan Zhao, Gershon Dublon, Nicholas Gillian, Artem Dementyev and Joseph A. Paradiso. 2015. EMI Spy: Harnessing electromagnetic interference for low-cost, rapid prototyping of proxemic interaction. In *Wearable and Implantable Body Sensor Networks* (BSN '15). IEEE, 6 pages. http://dx.doi.org/10.1109/BSN.2015.7299402