SurfaceSight: A New Spin on Touch, User, and Object Sensing for IoT Experiences

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

Figure 1. SurfaceSight enriches Internet-of-Things (IoT) experiences with touch, user, and object sensing. This is achieved by adding LIDAR to devices such as smart speakers (A). Next, we perform clustering and tracking (B), which unlocks novel interactive capabilities such as object recognition (C), touch input (D), and person tracking (E).

ABSTRACT

IoT appliances are gaining consumer traction, from smart thermostats to smart speakers. These devices generally have limited user interfaces, most often small buttons and touchscreens, or rely on voice control. Further, these devices know little about their surroundings – unaware of objects, people and activities happening around them. Consequently, interactions with these "smart" devices can be cumbersome and limited. We describe SurfaceSight, an approach that enriches IoT experiences with rich touch and object sensing, offering a complementary input channel and increased contextual awareness. For sensing, we incorporate LIDAR into the base of IoT devices, providing an expansive, ad hoc plane of sensing just above the surface on which devices rest. We can recognize and track a wide array of objects, including finger input and hand gestures. We can also track people and estimate which way they are facing. We evaluate the accuracy of these new capabilities and illustrate how they can be used to power novel and contextually-aware interactive experiences.

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300559

Author Keywords

Ubiquitous sensing; IoT; Smart Environments;

CCS Concepts

Human-centered computing~ Ubiquitous and mobile computing systems and tools.

ACM Reference format:
Gierad Laput and Chris Harrison. 2019. SurfaceSight: A New Spin on Touch, User, and Object Sensing for IoT Experiences. In *2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland, UK.* ACM, New York, NY, USA. 10 pages. https://doi.org/10.1145/3290605.3300559

1 INTRODUCTION

Small, internet-connected appliances are becoming increasingly common in homes and offices, forming a nascent, consumer-oriented "Internet of Things" (IoT). Product categories such as smart thermostats, light bulbs and speakers have shipped tens of millions of units in 2018 [9], with sales predicted to increase dramatically in the coming years.

Input on these devices tends to fall into one of three categories. First, we have products with extremely limited or no on-device input, which require an accessory physical remote or smartphone app for control (*e.g.,* Apple TV, Philips Hue bulbs). Second, and perhaps most pervasive at present, is for devices to offer some physical controls and/or a touchscreen for configuration and control (*e.g.,* Nest Thermostat, smart locks, smart refrigerators). Finally, there are "voice-first" interfaces [62] (*e.g.,* Google Home, Amazon Alexa, Apple HomePod). Regardless of the input modality, the user experience is generally recognized to be

cumbersome [49], with both small screens and voice interaction having well-studied HCI bottlenecks.

Another long-standing HCI research area and drawback of current generation consumer IoT devices is a limited awareness of context [1, 50]. An archetype of this interactive shortfall is a smart speaker sitting on a kitchen countertop, which does not know where it is, nor what is going on around it. As a consequence, the device cannot proactively assist a user in tasks or resolve even rudimentary ambiguities in user questions.

In this work, we investigate how the addition of low-cost LIDAR sensing into the base of consumer IoT devices can be used to unlock not only a complementary input channel (expansive, ad hoc touch input), but also object recognition and person tracking. Taken together, these capabilities significantly expand the interactive opportunities for this class of devices. We illustrate this utility through a set of functional example applications, and quantify the performance of main features in a multi-part study.

2 RELATED WORK

Our work intersects with several large bodies of HCI research, including ad hoc touch sensing, tracking of both objects and people in environments, and around-device interaction. We briefly review this expansive literature, focusing primarily on major methodological approaches. We then review other systems that have employed LIDAR for input and context sensing, as these are most similar to SurfaceSight in both function and operation.

2.1 Ad Hoc Touch Sensing

Research into enabling touch sensing on large, ad hoc surfaces (also referred to as "appropriated" interaction surfaces [20]) goes back at least two decades. By far, the most common approach is to use optical sensors, including infrared emitter-detector arrays [40], infrared cameras [23, 48], depth cameras [64, 66, 69] and thermal imaging [31]. Acoustic methods have also been well explored, using sensors placed at the periphery of a surface [22, 42] or centrally located [68]. Large scale capacitive sensing is also possible with some surface instrumentation (which can be hidden, *e.g.*, with paint), using discrete patches, tomographic imaging [75], and projective capacitive electrode matrices [76].

2.2 Sensing Objects in Environments

Many approaches for automatic object recognition have been explored in previous research. Typical methods involve direct object instrumentation, such as fiducial markers [25], acoustic barcodes [19], RFID tags [36], Bluetooth Low Energy tags and NFCs [16]. Although direct object instrumentation can be robust, it incurs installation and maintenance costs. Another approach is to sparsely instrument the environment with cameras [30, 32], radar [72], microphones [52], or worn sensors [26, 27, 38, 58, 61]. These minimally invasive approaches provide a practical alternative for object and human activity recognition.

2.3 Person Sensing and Tracking

Many types of systems – from energy efficient buildings [37] to virtual agents [54] – can benefit from knowledge of user presence, identification, and occupancy load. As such, many methods have been considered over many decades.

One approach is to have users carry a device such as a badge. Numerous systems with this configuration have been proposed, and they can be categorized as either active (*i.e.*, badge emits an identifier [21, 60]) or passive (*i.e.*, badge listens for environment signals [13, 44]). Badge-based sensing systems come in other forms, including radio frequency identification (RFID) tags [47], infrared proximity badges [60], microphones [21] and Bluetooth tags [53]. To avoid having to instrument users, researchers have also looked at methods including Doppler radar [46], RFID tracking [59] and co-opting WiFi signals [2, 45]. However, perhaps most ubiquitous are Pyroelectric Infrared (PIR) sensors, found in nearly all commercial motion detectors, which use the human body's black body radiation to detect motion in a scene. Also common are optical methods, including infrared (IR) proximity sensors [3] and pose-driven camera-based approaches [6].

2.4 Around-Device Interactions

Perhaps most similar to the overall scope of SurfaceSight is the subdomain of Around Device Interaction (ADI). This topic typically focuses on mobile and worn devices, and for capturing touch or gesture input. Several sensing principles have been explored, including acoustics [17, 41], hall-effect sensors [67], IR proximity sensors [5, 24, 29], electric field (EF) sensing [33, 77], magnetic field tracking [8, 18], and time-of-flight depth sensing [70]. Across all of these diverse approaches, the overarching goal is to increase input expressivity by leveraging the area around devices as an interaction surface or volume. Our technique adds to this rich body of prior work, adding a novel set of interaction modalities and contextual awareness capabilities.

2.5 LIDAR

Originally a portmanteau of light and radar, LIDAR uses the time-of-flight or parallax of laser light to perform range-finding. First developed in the 1960s, the initial high cost limited use to scientific and military applications. Today, LIDAR sensors can be purchased for under $10, for example, STMicroelectronics's VL53L0X [55]. The latter component is an example of a 1D sensor, able to sense distance along a

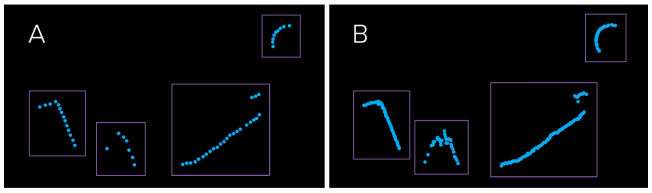

Figure 2. Each LIDAR rotational pass is slightly misaligned. We exploit this property by integrating data from multiple rotational passes. (A) shows a scene from a single pass, while (B) is integrated from 16 passes. Objects left to right: Mineral spirits can, user hand flat on surface, hammer, and bowl.

single axis. Electromechanical (most often spinning) 2D sensor units are also popular, starting under $100 in single-unit retail prices (*e.g.*, YDLIDAR X4 360° [71]). This is the type of sensor we use in SurfaceSight. Prices are likely to continue to fall (with quality increasing) due to economies of scale resulting from extensive LIDAR use in robotics and autonomous cars [35]. Solid state LIDAR and wide-angle depth cameras are likely to supersede electromechanical systems in the future; the interaction techniques we present in this paper should be immediately applicable, and likely enhanced with such improvements.

2.6 LIDAR in Interactive Systems

Although popular in many fields of research, LIDAR is surprisingly uncommon in HCI research. It is most commonly seen in Human-Robot Interaction (HRI) papers, where the robot uses LIDAR data to *e.g.*, track and approach people [34, 57, 74]. Of course, robots also use LIDAR for obstacle avoidance and recognition, which has similarities to our object recognition and tracking pipeline. Most similar to SurfaceSight are the *very* few systems that have used LIDAR for touch sensing. Amazingly, one of the very earliest ad hoc touch tracking systems, LaserWall [43, 56], first demonstrated in 1997, used spinning LIDAR operating parallel to a surface. Since then, we could only find one other paper, Digital Playgroundz [15], that has used such an approach. Further afield is Cassinelli *et al.* [7], which uses a steerable laser rangefinder to track a finger in mid-air.

3 IMPLEMENTATION

We now describe our full-stack implementation of SurfaceSight, from sensor hardware to event handling.

3.1 Hardware

For our proof-of-concept system, we use a Slamtech RPLIDAR A2 [50], which measures 7.6 cm wide and 4.1 cm tall. This is sufficiently compact so as to fit under most IoT devices (*e.g.*, speakers, thermostats). We suspend the unit upside down from an acrylic frame to bring the sensing plane to 6.0 mm above the host surface. In a commercial embodiment, we envision the sensor being fully integrated into the base of devices, with a strip of infrared translucent material that hides and protects the sensor.

The Slamtech RPLIDAR A2 can sense up to 12 m (15 cm minimum) with its Class 1 (eye-safe), 785nm (infrared) laser. Distance sensing is accurate to within ±3 mm at distances under 3 meters. We modified the device driver to rotate at maximum speed (12 Hz) and maximum sampling rate (4 kHz), providing an angular resolution of ~1.1°.

3.2 Scene Subsampling

Each LIDAR rotational pass is slightly misaligned, offering the ability to subsample object contours by integrating data from multiple rotational passes (Figure 2). This presents an interesting tradeoff: on one end of the spectrum, we can capture sparse contours that update as quickly as a single rotation (Figure 2A). On the other end, we can integrate many rotational passes to collect high quality, dense contours (Figure 2B), which also permits the capture of smaller objects at longer distances. This, of course, incurs a non-trivial time penalty, and also leaves behind "ghost" points whenever an object is moved.

Fortunately, we can achieve the best of both worlds by maintaining two independent polar point cloud buffers, with different integration periods. First is our "finger" buffer, which integrates five rotations (*i.e.*, 2.4 FPS) with an effective angular resolution of ~0.5°. We found this integration period offered the best balance between robustly capturing small fingers, while still offering interactive framerate. Our second, "object" buffer, integrates 16 rotational passes (*i.e.*, 0.75 FPS) for an effective angular resolution of ~0.2°, which we found strikes a balance between update rate and object contour quality.

3.3 Clustering

We cluster our point clouds using a variant of the adaptive breakpoint detection (ABD) scheme introduced by Borges *et al.* [4]. Two points are part of the same cluster if their

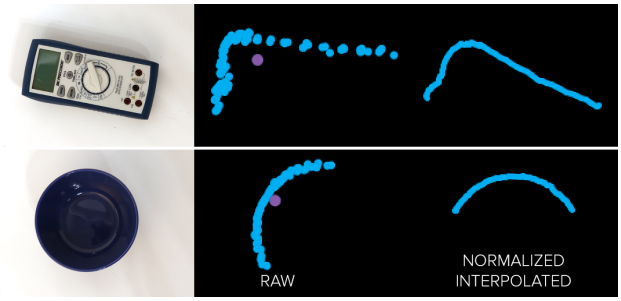

Figure 3. For each cluster, we transform all points into a local coordinate system, rotate, and resample them for feature extraction.

Euclidean distance falls below a dynamic, distance-based threshold, defined by the following formula:

$$t_{breakpoint} = a * D^2 + b * D + c$$

where D is the distance in mm, and a, b, and c are empirically determined coefficients. We computed these values (a=5e-5, b=0.048, and c=18.46) by capturing pilot data in four commonplace environments with existing objects present. The output of clustering is an array of objects, each containing a series of constituent points.

3.4 Feature Extraction

Once individual points have been grouped into a single cluster, we transform all points into a local coordinate system, rotate the point cloud to align with the 0°-axis of the sensor, and resample the contour into a 64-point path (Figure 3, right). This helps homogenize object contours into a distance-from-sensor and rotation-invariant form.

We then generate a series of cluster-level features that characterizes objects for recognition. Specifically, we compute the following features for each cluster: area of bounding box, real world length of path, relative angle between consecutive points, and angles between each point relative to the path centroid. Next, we draw a line between the first and last point in a path, and compute the residuals for all intermediate points, from which we derive seven statistical values: min, max, mean, sum, standard deviation, range, and root-mean squared (RMS). Finally, we take every fourth residual and compute its ratio against all others.

3.5 Object Classification & Viewpoint Invariance

Before classification of clusters can occur, a model must be trained on objects of interest. To achieve viewpoint independence, we capture training data from many viewpoints. We maintain a database of all previously seen object contours (featurized), which allows us to compute an incoming contour's nearest neighbor (linear distance function). In essence, these viewpoints are treated as independent objects, that happen map to a single object class. If the contour is below a match threshold, it is simply ignored. If one or more matches are found, the contour proceeds to object classification. Rather than use the nearest neighbor result, we found better results when using a random forest classifier (batch size=100, max depth=unlimited, default parameters on Weka 11).

3.6 Cluster Tracking

Feature computation and classification occurs once, when a cluster is first formed. From that point on, the cluster is tracked across frames, and the classification result is carried forward. A persistent cluster ID is also important for tracking finger stokes and detecting gestures. For tracking, we use a greedy, Euclidean distance pairwise matching approach with a distance threshold. Although simple, it works well in practice. We maintain a movement history of 1.0 seconds for all clusters, which provides trajectory information. Our tracking pipeline is also responsible for generating *on-down*, *on-move* and *on-lift* events that trigger application-level interactive functions.

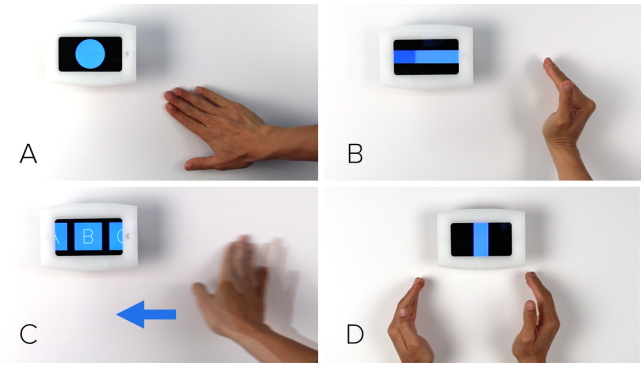

Figure 4. SurfaceSight enables touch and gesture recognition. Here, we show SurfaceSight operating on a wall enabling buttons (A), sliders (B), swipe carousels (C), and two-handed gestures (D).

3.7 Touch Input and Gesture Recognition

Recognition of finger inputs is handled identically to other objects (as it has a distinctive shape and size), except that we use our high framerate "finger" buffer for tracking. Positional tracking immediately enables virtual widgets, such as buttons on ad hoc surfaces (Figure 4A & B). As noted above, we maintain a one-second movement history for every cluster. In the case of finger inputs, we use this motion vector for stroke gesture recognition. We support six unistroke gestures: *up, down, left, and right* swipes (Figure 4C), as well as *clockwise*, and *counter-clockwise* rotations. For this, we used the $1 recognizer [65] by Wobbrock *et al.* In addition to motion gestures, SurfaceSight can also recognize ten static hand poses (Figures 4D and 12): *point, all fingers together, flat palm, fist, wall, corner, stop, 'V', circle,* and *heart.* As these are whole-hand shapes, as opposed to motions, we register these contours in our system in the exact same manner as physical objects.

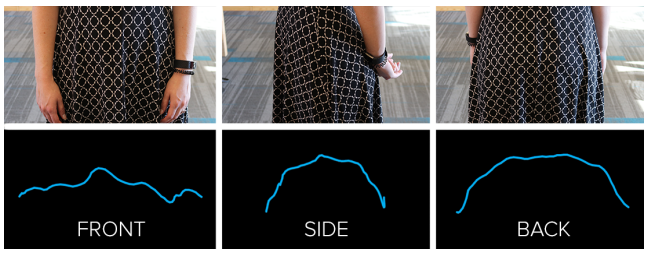

Figure 5. Our system can detect people and recognize different sides of bodies (example contours in blue).

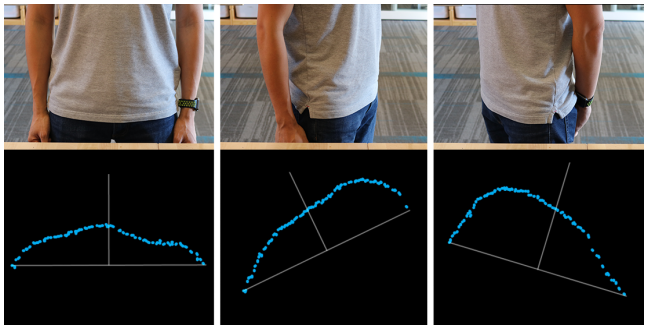

Figure 6. If SurfaceSight detects that a person is facing forwards, it also computes a body angle estimation. This extra contextual information could be used to e.g., enable voice interaction without a keyword when the user is sufficiently close and facing a device.

3.8 Person Tracking and Body Angle Estimation

Finally, SurfaceSight can also recognize people as another special object class. Human contours are large, move in characteristic trajectories, and have different contours from most inanimate objects. We created three subclasses: body *front, back,* and *side* (Figure 5), which have unique contours. If we detect that a person is facing *front,* we perform an extra step to estimate which angle they are facing. For this, we create a line between the first and last points in the cluster, and project an orthogonal vector from the midpoint (Figure 6, bottom row). From this data, it is also possible to attribute touch points to a person, as shown in Medusa [3].

3.9 Defining the Interactive Area

The planar sensing offered by LIDAR can easily identify concave adjoining surfaces, such as the transition from a countertop to backsplash, or desk to wall. However, convex discontinuities, such as the outer edge of countertop or desk, are invisible to the sensor. This edge represents an important functional boundary between "human" space (floors) and "object" space (raised surfaces). For example, you are likely to see a cross-section of a human torso out in a room, but not on a countertop. While it may be possible for the system to learn this boundary automatically, by tracking where objects appear over time, we leave this to future work. Instead, we built a rapid initialization procedure, where users are requested to touch the outer perimeter of a work surface, on which we compute a convex hull. It is also possible to specify a fixed interactive radius, *e.g.,* one meter.

4 EXAMPLE APPLICATIONS

SurfaceSight enables six input modalities: virtual widgets, static hand poses, finger motion gestures, object recognition, people tracking, and body angle estimation. These fundamental capabilities can be incorporated into a wide variety of end user applications. In this section we offer five example applications to illustrate potential uses, for both walls and horizontal surfaces. Please also see Video Figure.

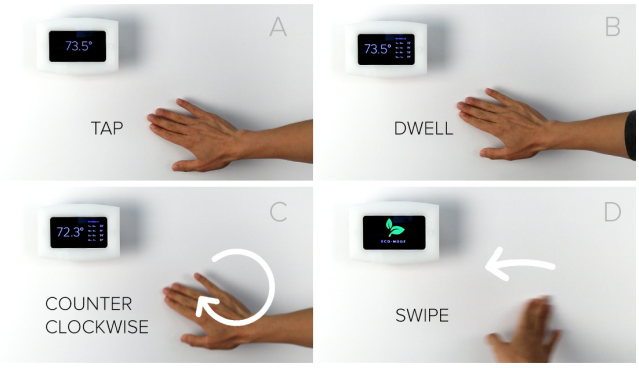

Figure 7. Thermostat demo application. Tapping the wall wakes the device (A), and a dwell reveals more details (B). Motion gestures trigger specific commands, such as fined-grained temperature adjustment (C) or swiping between different temperature presets (D).

4.1 Thermostat

We created a SurfaceSight-enhanced thermostat demo that responds to finger touches within a 1-meter radius (Figure 7). Picture frames, people leaning against the wall, and similar non-finger objects are ignored. Tapping the wall wakes the device to display the current temperature, whereas a longer dwell reveals a more detailed HVAC schedule. Clockwise and counterclockwise circling motions adjust

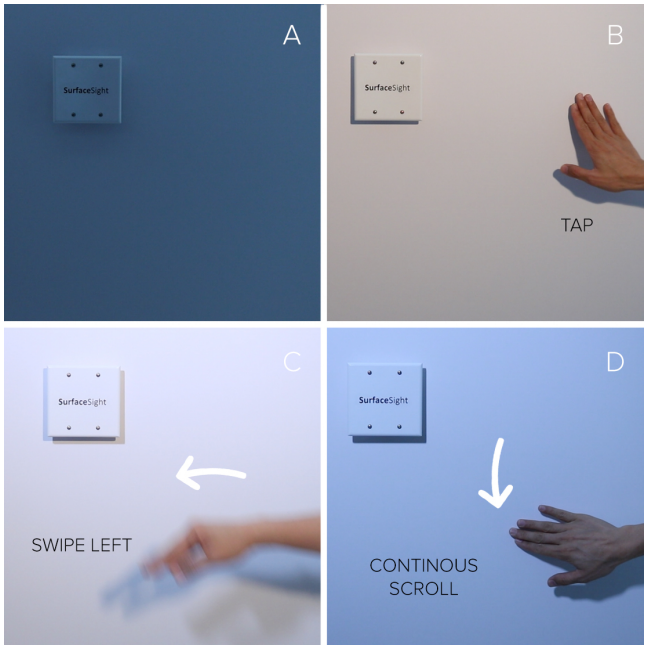

Figure 8. Light switch demo application. Tapping the wall turns the light on (B), swipes reveal different lighting modes (C). Continuous up or down scrolling adjusts lighting illumination levels (D).

the desired temperature up or down. Finally, swipes to the left and right navigate between different modes, such as eco mode, away from home, fast heat, and fast cool.

4.2 Light switch

As a second wall demo, we created an augmented light switch (Figure 8). Instead of a physical toggle button, all interactions are driven through touches to the wall. A tap is used to toggle lights on or off. Sliding up and down functions as a dimmer control. Finally, we detect left and right swipes to move between lighting presets, such as incandescent, daylight, evening, and theater modes.

4.3 Recipe Helper

We augmented an Amazon Alexa (Figure 9), which we can programmatically control through its Alexa Skills Kit API. We situated this on a kitchen countertop and built a recipe app demo that can recognize common kitchenware, including mixing bowls, mortar, chopping board, and measuring cups of various sizes. If the recipe app requests an object as part of a recipe step (*e.g.,* "retrieve the mixing bowl"), it automatically advances to the next instruction once that item is placed on the surface. Likewise, questions with an ambiguous object are assumed to be the last item that appeared or was moved by the user. As a demo, we implemented a "how many *[units]* are in this?" command. In our Video Figure, the user asks "how many ounces in this?" after putting down a measuring cup. Finally, swiping left or right allows the user to rapidly navigate between recipe steps, including the ability to replay the current step.

Figure 9. Recipe helper demo application. When a recipe is loaded, the system asks the user to retrieve necessary items (A), moving to the next step automatically upon detection (B). Contextual questions such as "how many ounces are in this" after putting down a measuring cup are possible (C). Swipe gestures move between recipe steps (D & E). When a user is finished with a step (*e.g.,* mortar and pestle lifted from surface), the recipe can automatically advance (F).

4.4 Music Player

We created a music player demo using an instrumented Google Home. This scans for phones resting nearby on the *same* surface (Figure 10), which is interpreted to be an explicit action by a user to connect the two devices. This is in contrast to *e.g.,* automatic Bluetooth pairing, which might occur when the device is in the pocket of a nearby user. Once connected, music can be controlled by using the table's surface: tap to pause/play, left and right swipes to move between songs, left and right continuous motions to scrub inside of a song, and sliding up and down to control volume. As noted earlier, smart speakers have trouble with

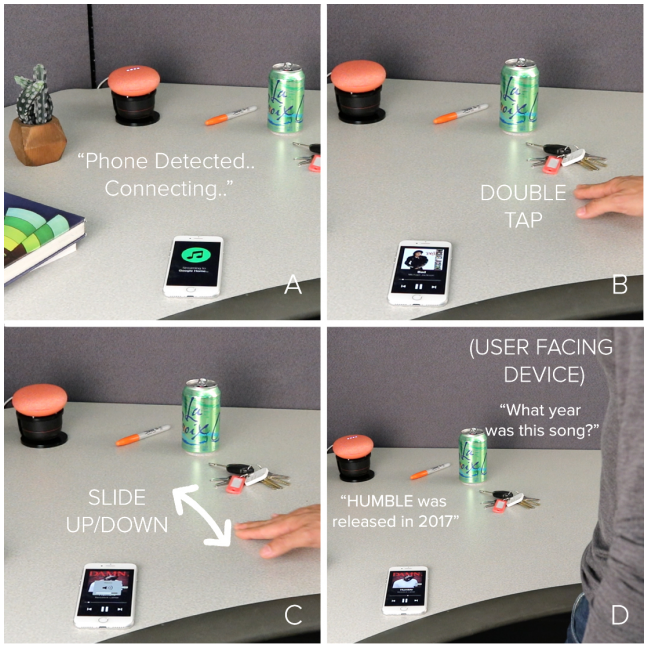

Figure 10. Music player demo application. When the smart speaker detects a smartphone on the table (A), audio streaming is initialized. Gestures such as double tap (B) and continuous scrolling (C) control audio playback and volume. The system also tracks the user's body angle, lowering the playback volume if a user turns to the device in anticipation of a (potentially wakeword-less) spoken command.

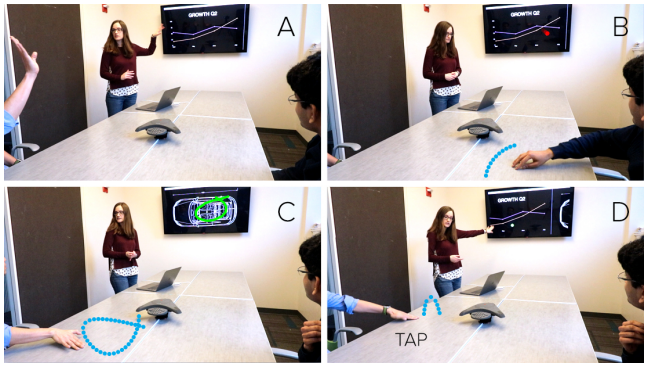

Figure 11. Conference room speakerphone augmented with SurfaceSight (A). Meeting participants can summon a cursor when touching the table in front of them like a trackpad (B). Tapping the table toggles between cursor and draw modes (C). Participants can also move through the slide deck by tapping to their left and right (D).

spoken input when playing content. In our demo, the music volume is momentarily halved when a user turns to face the Google Home, in anticipation of a spoken command.

4.5 Conference Room

Finally, we augmented a conference room speakerphone with SurfaceSight, allowing the conference table to become a collaborative tool (Figure 11). First, the system localizes meeting participants, and initializes virtual, interactive trackpads in front of each user. Touching this interactive region triggers a colored cursor to appear on a shared presentation (*e.g.*, projected). Tapping the table toggles between cursor and draw modes, allowing users to point at or highlight information on the slides. Users can also collaboratively control the slide deck by tapping to their left or right extremes.

5 EVALUATION

In our evaluation, we sought to quantify four key questions: 1) What is the system's touch sensing accuracy? 2) How well does the system recognize static and dynamic hand gestures? 3) What is the accuracy of object detection across several commonplace use environments? 4) How accurate is person detection and body angle estimation?

We recruited 14 participants (4 female, mean age 29.2), from a public participant pool. We collected self-reported height and weight in our demographics form, which allowed us to estimate BMI (NIH formula). We had 8 normal weight, 4 overweight and 2 obese participants. We found no correlation in accuracy across any of our demographics and thus we only report combined results. Our first four studies were conducted on a generic wooden table, offering an interaction surface 90 × 210 cm. We placed our SurfaceSight prototype opposite participants, centered on the long edge of the table. To facilitate data capture, we installed a projector above the table in order to render automated visual instructions and targets for participants to follow (calibrated to SurfaceSight's coordinate system). The study lasted one hour and paid $10.

5.1 Study #1: Touch Sensing

To assess touch sensing accuracy, we designed a target acquisition task, where participants were asked to touch the center of a crosshair (14×6 grid, 15 cm spacing, 84 positions total, random order). Users were allowed to use either hand interchangeably, and they were not required to remove accessories, jewelry, or make any clothing adjustments. For each trial, we measured the error between the crosshair position and our touch tracker's output (*i.e.*, cluster centroid). We ran data collection twice, once with participants using a single finger for input, and a second time with multiple fingers held together.

Across 14 users and 2,300 cumulative touch trials, our system achieved a mean touch Euclidian distance error of ±1.60 cm (SD=0.7). We found a linear relationship between touch error and the target's distance from the sensor. There was no significant difference in spatial accuracy when using one or more fingers for input. However, in the single finger input condition, 9.2% (SD=5.9) of trials were missed on average (no touches were missed in the multiple finger input condition). Of the missed trials, 97% were on targets greater than 0.8m away, suggesting a drop in performance at longer ranges especially for small interactors such as a finger.

5.2 Study #2: Motion Gestures

We also investigated how well SurfaceSight could detect dynamic, motion gestures. For this task, we selected six directional swipes common in the literature: left, right, up, down, clockwise, and counterclockwise. Participants performed each gesture twice (in random order), on a 2×3 grid

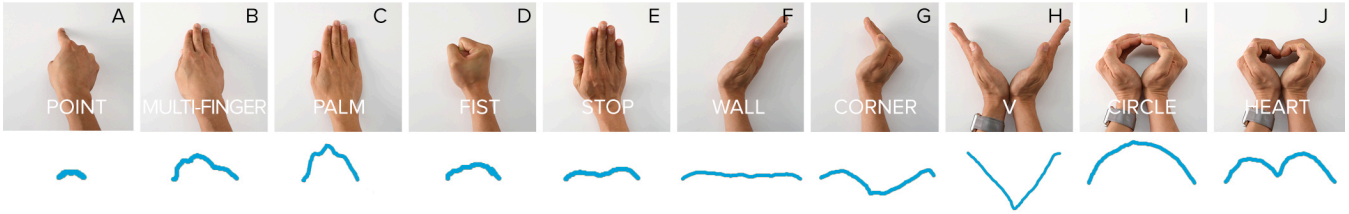

Figure 12. The ten static hand poses used in the evaluation and their corresponding contours (variable scale).

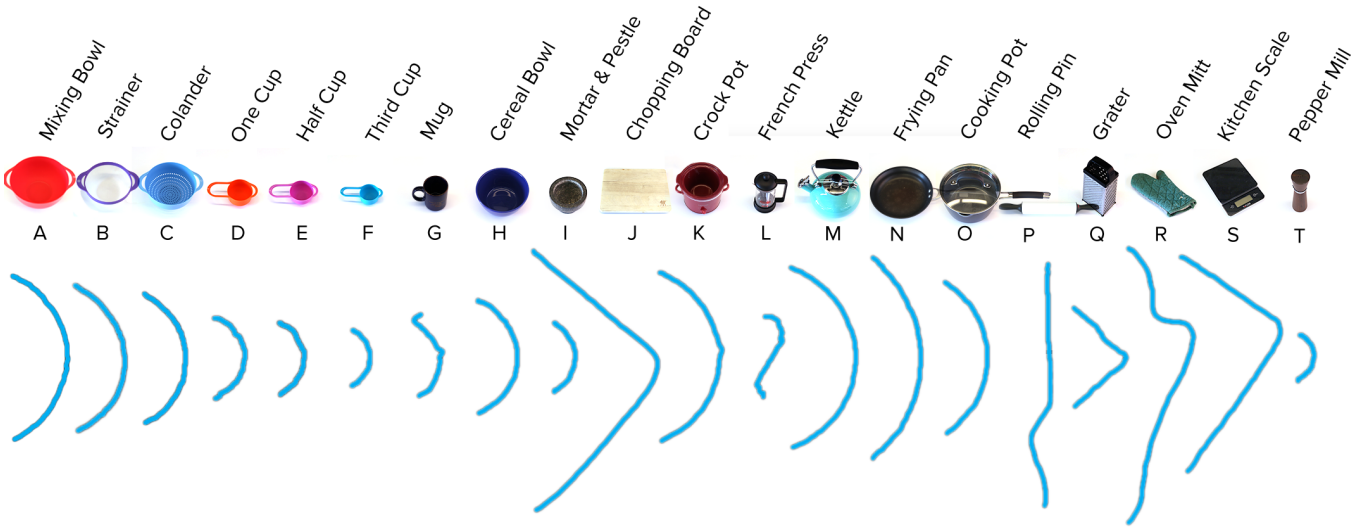

Figure 13. Kitchen object set used in the recognition evaluation. Here, we show the name, photo (not to scale), and ID of each object, along with their normalized-interpolated contour in blue (absolute scale, vertically oriented along principle axis).

(same table as Study 1). Similar to our previous study, users were free to use either hand. In total, this procedure yielded 6 gestures × 2 repeats × 6 grid locations × 14 participants = 1008 trials. Gesture detection was performed live.

Across our 14 participants, SurfaceSight was able to correctly capture and classify motion gestures with an accuracy of 97.3% (SD=1.7), with most errors occurring at farther distances (consistent with the previous study result).

5.3 Study #3: Static Hand Postures

Beyond motion gestures, we also sought to evaluate how well our system can detect *static* hand poses. For this task, we asked users to perform ten static hand poses, which included common single- and two-handed gestures (see *e.g.*, [12, 63]), depicted in Figure 12. This study was segmented into two parts: training and testing. In the training phase, users performed all ten gestures in random locations across the table's surface, and an experimenter segmented data to train a machine learning model (see Implementation section). In the testing phase, a model was trained using the collected data and gesture classification was performed live by the system. Similar to the previous study, users were asked to perform all ten gestures (random order) on a 2×3 grid. At each grid location, participants performed 10 static hand gestures = 60 trials total.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	80%	3%	0%	0%	0%	0%	0%	7%	0%	0%	3%	0%	3%	0%	3%	0%	0%	0%	0%	0%
B	3%	80%	0%	0%	0%	0%	0%	0%	0%	0%	7%	0%	0%	0%	10%	0%	0%	0%	0%	0%
C	0%	0%	90%	0%	0%	0%	0%	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
D	0%	0%	0%	97%	0%	0%	0%	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
E	0%	0%	0%	10%	90%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
F	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
G	0%	0%	0%	7%	0%	0%	90%	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
H	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
I	0%	0%	0%	3%	3%	0%	10%	0%	83%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
J	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
K	7%	3%	0%	0%	0%	0%	0%	3%	0%	87%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
L	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	97%	0%	0%	0%	0%	0%	0%	0%	0%	3%
M	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	97%	0%	0%	0%	0%	0%	0%	0%	0%
N	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
O	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
P	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
Q	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	97%	0%	0%	0%	0%
R	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
S	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	0%	97%	0%
T	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

Figure 14. Confusion matrix for kitchen objects. Object keys can be found in Figure 13. Mean accuracy across all 20 objects is 94.2%.

Across 14 users and 840 cumulative gesture trials, our system was able to capture and classify static hand gestures with an accuracy of 96.0% (SD=3.0). We found no significant effect of distance, likely because hand gestures have much larger area than single fingers.

5.4 Study #4: Body Angle

Next, we sought to evaluate how well our system can detect a person and their body angle. For this study, we selected seven equally spaced locations around the left, right, and bottom edges of our test table. To request a location, we projected a 0.5 m semicircle on the table's surface. We then projected a line radiating from the center of the circle (random angle ±15°), indicating which way a participant should orient their body. We then recorded the angular difference between the target line and our system's predicted body angle. We repeated this process three times per location, for a total of 21 trials per participant. Similar to the previous studies, all predictions were performed live.

Across 14 users and 294 trials, person localization was 100% accurate (*i.e.*, the system recognized a person was standing at the requested spot). For these trials, our system predicted body angle with a mean angular error of 3.0°

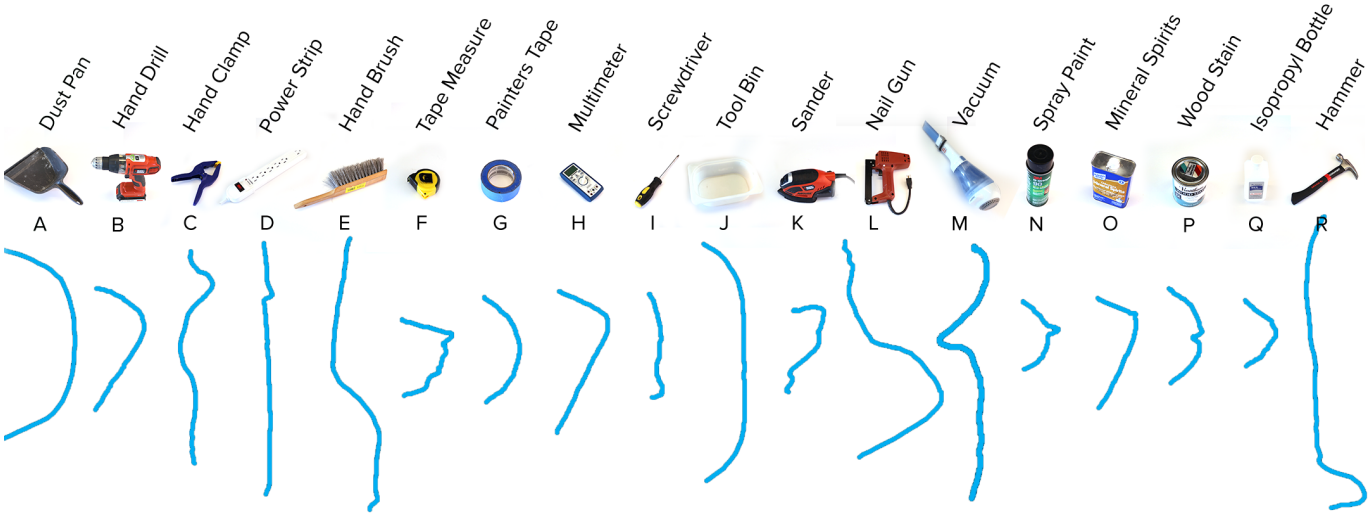

Figure 15. Workshop object set. Similar to Figure 13, we show the name, photo (not to scale) and ID of each object, along with the normalized-interpolated contour in blue (absolute scale, oriented vertically along principle axis).

(SD=3.7). We found no significant difference between location/distance and body angle estimation accuracy.

5.5 Study #5: Object Recognition

In our final study (no users involved), we assessed how well SurfaceSight can recognize objects based solely on their contours. For this, we collected 38 everyday objects from our building and split them into two functional categories: kitchen and workshop. While we did not reject any objects due to identical contours (we did not encounter any), we did curate this collection according to two criteria: 1) Diversity - we wished to show the breadth of our system's recognition, and 2) Utility - we sought items that could be put together into interesting and illustrative example apps.

Similar to the previous study procedures, we segmented this study into training and testing phases. In the training phase, we captured data for each object (different distances and angles), roughly 1000 instances per object. In the testing phase, we trained one model per category, and we performed live object classification. In each trial, a random position, angle and object was requested (via projector). In total, we captured 1,140 trials (38 objects × 30 random positions/angles).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
A	90%	3%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	3%
B	0%	90%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	0%	7%	0%	0%	0%
C	0%	0%	90%	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	3%
D	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
E	0%	0%	0%	0%	93%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	0%
F	0%	0%	0%	0%	0%	90%	0%	0%	0%	0%	0%	0%	3%	3%	0%	3%	0%	0%
G	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
H	0%	0%	0%	0%	0%	0%	0%	93%	0%	0%	0%	0%	3%	0%	0%	0%	0%	3%
I	0%	3%	0%	0%	0%	0%	0%	0%	93%	0%	0%	0%	0%	0%	0%	0%	3%	0%
J	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
K	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	93%	0%	0%	0%	0%	0%	0%	3%
L	0%	7%	0%	0%	0%	0%	0%	0%	3%	0%	7%	83%	0%	0%	0%	0%	0%	0%
M	0%	0%	0%	0%	0%	0%	13%	0%	0%	0%	0%	0%	83%	0%	0%	0%	3%	0%
N	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	93%	0%	0%	3%	0%
O	0%	3%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	0%	93%	0%	0%	0%
P	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	0%	0%	97%	0%	0%
Q	0%	0%	0%	0%	0%	7%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	93%	0%
R	0%	13%	0%	0%	0%	0%	0%	0%	3%	0%	0%	3%	0%	0%	0%	0%	3%	77%

Figure 16. Confusion matrix for workshop objects. Objects keys are in Figure 15. Accuracy across 18 objects is 91.8%.

Across all trials, SurfaceSight achieved an overall object recognition accuracy of 93.1% (SD=6.6). Recognition accuracy was 94.2% (SD=6.9) for kitchen objects, and 91.8% (SD=27.4) for workshop objects. These results highlight SurfaceSight's ability to robustly infer objects based solely on their contours, creating opportunities for imbuing IoT devices with contextual awareness that is difficult or impossible to achieve with existing methods.

6 FAILURE MODES & LIMITATIONS

The biggest limitation of our system, and LIDAR in general, is occlusion. Everyday surfaces such as kitchen countertops, dining tables, and even walls are messy and rife with clutter. Relying solely on line-of-sight means that certain events of interest will be missed. We offer a few ways to address this limitation. First, designating the extents of an interactive area could enable the system to automatically monitor occlusion and provide user feedback (*i.e.*, a clutter detector). Second, we can leverage deep learning generative models (*e.g.*, GANs [14, 73]) to "fill-in" gaps (*e.g.*, when an object is partially blocked). Finally, we can take advantage of motion trajectories and perform tracking prediction (*e.g.*, as shown by Ess *et al.* [10]) to mitigate occlusion effects.

We are also limited by sensing geometry. LIDAR works best on level surfaces and its data is inherently planar.

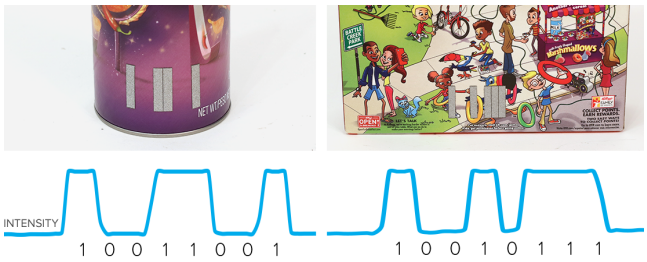

Figure 17. We can tag objects with retro-reflective barcodes, increasing the set of objects that can be detected.

Further, our LIDAR-based sensing approach is also constrained in the types of objects it can detect. We are subject to collisions in object contours (*i.e.,* similar shaped, but different objects) and objects that do not reflect infrared. For example, our system is unable to detect transparent materials (*e.g.,* glass) or objects with highly specular surfaces (*e.g.,* mirror finishes). One way to overcome this is to attach "tags" to objects, allowing them to reflect infrared. It is also possible to embed data into these tags, similar to a barcode. Figure 17 shows the signal received from two 8-bit tags we created. Such a method could also be used to disambiguate objects with identical contours (though unexpectedly rare).

It is also possible to leverage complementary sensors – such as panoramic cameras, depth cameras and sonar – to make SurfaceSight more capable and accurate. Of course, these sensors have their own limitations. For example, cameras offer high-fidelity visual information, but must deal with variable lighting and scale, generally require more compute to process, and have privacy implications in *e.g.,* the home. Depth cameras could conceivably be made with 360° sensing ability, or several units could be used in concert, and operate much like LIDAR. Sonar is a cheaper alternative, but it is imprecise for the types of applications we envision. LIDAR, in contrast, is robust across lighting conditions, works at long distances, and provides very accurate contour data in a convenient form (*i.e.,* a planar point cloud with real-world units).

7 CONCLUSION

SurfaceSight, using LIDAR, enables expressive input and enhanced contextual awareness, including the detection of objects, finger touches, hand gestures, people tracking, and body angle estimation. Our evaluations suggest immediate feasibility, and our example applications illustrate how SurfaceSight can be used to power novel, contextually-aware interactive experiences.

ACKNOWLEDGEMENTS

This work was generously supported by Intel, as well as funds from the Packard Foundation, Sloan Foundation and Google Ph.D. Fellowship. We specifically thank our Intel mentors, Carl Marshall and Moh Haghighat, for their support and encouragement. We also thank Evi Bernitsas for her early work on sensor integration.

REFERENCES

1 Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. 1999. Towards a Better Understanding of Context and Context-Awareness. In Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing (HUC '99). Springer-Verlag, London, UK, 304-307.

2 Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. ACM Trans. Graph. 34, 6, Article 219 (October 2015), 13 pages. DOI: https://doi.org/10.1145/2816795.2818072

3 Michelle Annett, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2011. Medusa: a proximity-aware multi-touch tabletop. In Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11). ACM, New York, NY, USA, 337-346. DOI: https://doi.org/10.1145/2047196.2047240

4 Geovany Araujo Borges and Marie-José Aldon. Line extraction in 2D range images for mobile robotics. Journal of Intelligent & Robotic Systems, 40(3):267– 297, 2004.

5 Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-"touch" interaction around small devices. In Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08). ACM, New York, NY, USA, 201-204. DOI: https://doi.org/10.1145/1449715.1449746

6 Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime multi-person 2d pose estimation using part affinity fields." arXiv preprint arXiv:1611.08050 (2016).

7 Álvaro Cassinelli, Stéphane Perrin, and Masatoshi Ishikawa. 2005. Smart laser-scanner for 3D human-machine interface. In CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05). ACM, New York, NY, USA, 1138-1139. DOI=http://dx.doi.org/10.1145/1056808.1056851

8 Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D input using two magnetic sensors. In Proceedings of the 26th annual ACM symposium on User interface software and technology (UIST '13). ACM, New York, NY, USA, 237-244. DOI: http://dx.doi.org/10.1145/2501988.2502035

9 Louis Columbus. "2017 Roundup of Internet of Things Forecasts". Forbes, Dec 10, 2017.

10 Andreas Ess and Konrad Schindler and Bastian Leibe and Luc Van Gool. 2009. Improved Multi-Person Tracking with Active Occlusion Handling. In Proceedings of IEEE ICRA '09.

11 Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 4th Ed., 2016.

12 Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. 2009. ShadowGuides: visualizations for in-situ learning of multi-touch and whole-hand gestures. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09). ACM, New York, NY, USA, 165-172. DOI: https://doi.org/10.1145/1731903.1731935

13 Mehdi Golestanian and Christian Poellabauer. 2016. Indoor localization using multi-range beaconing: poster. In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '16). ACM, New York, NY, USA, 397-398. DOI: https://doi.org/10.1145/2942358.2942414

14 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, nd Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In Advances in Neural Information Processing Systems 27. 2672--2680.

15 Dan Gregor, Ondrej Prucha, Jakub Rocek, and Josef Kortan. 2017. Digital playgroundz. In ACM SIGGRAPH 2017 VR Village (SIGGRAPH '17). ACM, New York, NY, USA, Article 4, 2 pages. DOI: https://doi.org/10.1145/3089269.3089288

16 Tobias Grosse-Puppendahl, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner

Wichert, and Arjan Kuijper. 2014. Capacitive near-field communication for ubiquitous interaction and perception. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14). ACM, New York, NY, USA, 231-242. DOI: https://doi.org/10.1145/2632048.2632053

17 Teng Han, Khalad Hasan, Keisuke Nakamura, Randy Gomez, and Pourang Irani. 2017. SoundCraft: Enabling Spatial Interactions on Smartwatches using Hand Generated Acoustics. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17). ACM, New York, NY, USA, 579-591. DOI: https://doi.org/10.1145/3126594.3126612

18 Chris Harrison and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09). ACM, New York, NY, USA, 121-124. DOI: https://doi.org/10.1145/1622176.1622199

19 Chris Harrison, Robert Xiao, and Scott Hudson. 2012. Acoustic barcodes: passive, durable and inexpensive notched identification tags. In Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12). ACM, New York, NY, USA, 563-568. DOI: https://doi.org/10.1145/2380116.2380187

20 Chris Harrison. Appropriated Interaction Surfaces. IEEE Computer Magazine, June 2010, 43(6). IEEE, Washington, D.C. 86-89.

21 Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. 1999. The anatomy of a context-aware application. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99). ACM, New York, NY, USA, 59-68. DOI=http://dx.doi.org/10.1145/313451.313476

22 Hiroshi Ishii, Craig Wisneski, Julian Orbanes, Ben Chun, and Joe Paradiso. 1999. PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99). ACM, New York, NY, USA, 394-401. DOI: http://dx.doi.org/10.1145/302979.303115

23 Hideki Koike, Motoki Kobayashi. Integrating paper and digital information on EnhancedDesk: a method for realtime finger tracking on an augmented desk system. ACM Trans. on Computer-Human Inter., 8(4), 307-322.

24 Sven Kratz and Michael Rohs. 2009. HoverFlow: expanding the design space of around-device interaction. In Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09). ACM, New York, NY, USA, Article 4, 8 pages. DOI=http://dx.doi.org/10.1145/1613858.1613864

25 Beth M. Lange, Mark A. Jones, and James L. Meyers. 1998. Insight lab: an immersive team environment linking paper, displays, and data. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98), ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 550-557. DOI=http://dx.doi.org/10.1145/274644.274718

26 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. DOI: http://dx.doi.org/10.1145/2807442.2807481

27 Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18). ACM, New York, NY, USA, 213-224. DOI: https://doi.org/10.1145/3242587.3242609

28 Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). ACM, New York, NY, USA, 321-333. DOI: http://dx.doi.org/10.1145/2984511.2984582

29 Gierad Laput, Robert Xiao, Xiang 'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. In Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14). ACM, New York, NY, USA, 389-394. DOI: https://doi.org/10.1145/2642918.2647356

30 Gierad Laput, Walter S. Lasecki, Jason Wiese, Robert Xiao, Jeffrey P. Bigham, and Chris Harrison. 2015. Zensors: Adaptive, Rapidly Deployable, Human-Intelligent Sensor Feeds. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 1935-1944. DOI=http://dx.doi.org/10.1145/2702123.2702416

31 Eric Larson, Gabe Cohn, Sidhant Gupta, Xiaofeng Ren, Beverly Harrison, Dieter Fox, and Shwetak Patel. 2011. HeatWave: thermal imaging for surface user interaction. In Proceedings of the SIGCHI Conference on Human Fac-tors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2565-2574. DOI: https://doi.org/10.1145/1978942.1979317

32 Walter S. Lasecki, Young Chol Song, Henry Kautz, and Jeffrey P. Bigham. 2013. Real-time crowd labeling for deployable activity recognition. In Proceedings of the 2013 conference on Computer supported cooperative work (CSCW '13). ACM, New York, NY, USA, 1203-1212. DOI: https://doi.org/10.1145/2441776.2441912

33 Mathieu Le Goc, Stuart Taylor, Shahram Izadi, and Cem Keskin. 2014. A low-cost transparent electric field sensor for 3d interaction on mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 3167-3170. DOI: https://doi.org/10.1145/2556288.2557331

34 Eui-Jung Jung, Jae Hon Lee, Byung-Ju Yi, Jooyoung Park, Shin'ichi Yuta and Si-Tae Noh, "Development of a Laser-Range-Finder-Based Human Tracking and Control Algorithm for a Marathoner Service Robot," in IEEE/ASME Transactions on Mechatronics, vol. 19, no. 6, pp. 1963-1976, Dec. 2014. DOI: 10.1109/TMECH.2013.2294180

35 Timothy Lee. Why experts believe cheaper, better lidar is right around the corner. Are Technica, January 1, 2018. Retrieved September 12, 2018. https://arstechnica.com/cars/2018/01/driving-around-without-a-driver-lidar-technology-explained/

36 Hanchuan Li, Can Ye, and Alanson P. Sample. 2015. IDSense: A Human Object Interaction Detection System Based on Passive UHF RFID. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). ACM, New York, NY, USA, 2555-2564. DOI: https://doi.org/10.1145/2702123.2702178

37 Zhi Liu, Jie Zhang and L. Geng, "An Intelligent Building Occupancy Detection System Based on Sparse Auto-Encoder," 2017 IEEE Winter Applications of Computer Vision Workshops (WACVW), Santa Rosa, CA, 2017, pp. 17-22.

38 Takuya Maekawa, Yasue Kishino, Yasushi Sakurai, and Takayuki Suyama. 2013. Activity recognition with hand-worn magnetic sensors. Personal Ubiquitous Comput. 17, 6 (August 2013), 1085-1094. DOI=http://dx.doi.org/10.1007/s00779-012-0556-8

39 Trinh Minh Tri Do, Kyriaki Kalimeri, Bruno Lepri, Fabio Pianesi, and Daniel Gatica-Perez. 2013. Inferring social activities with mobile sensor networks. In Proceedings of the 15th ACM on International conference on multimodal interaction (ICMI '13). ACM, New York, NY, USA, 405-412. DOI: https://doi.org/10.1145/2522848.2522894

40 Jon Moeller and Andruid Kerne. 2012. ZeroTouch: an optical multi-touch and free-air interaction architecture. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2165-2174. DOI: http://dx.doi.org/10.1145/2207676.2208368

41 Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 1515-1525. DOI: https://doi.org/10.1145/2858036.2858580

42 Joseph Paradiso, Che King Leo, Nisha Checka and Kaiyuh Hsiao. Passive acoustic sensing for tracking knocks atop large interactive displays. In Proc. IEEE Sensors '02. 521-527.

43 Joseph Paradiso, Kaiyuh Hsiao, Joshua Strickon, Joshua Lifton and A. Adler. Sensor Systems for Interactive Surfaces. IBM Systems Journal, Volume 39, Nos. 3 & 4, October 2000, pp. 892-914.

44 Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. 2000. The Cricket location-support system. In Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom '00). ACM, New York, NY, USA, 32-43. DOI=http://dx.doi.org/10.1145/345910.345917

45 Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In

Proceedings of the 19th annual international conference on Mobile computing & networking (MobiCom '13). ACM, New York, NY, USA, 27-38. DOI: http://dx.doi.org/10.1145/2500423.2500436

46 Tauhidur Rahman, Alexander T. Adams, Ruth Vinisha Ravichandran, Mi Zhang, Shwetak N. Patel, Julie A. Kientz, and Tanzeem Choudhury. 2015. DoppleSleep: a contactless unobtrusive sleep sensing system using short-range Doppler radar. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15). ACM, New York, NY, USA, 39-50. DOI: https://doi.org/10.1145/2750858.2804280

47 Lenin Ravindranath, Venkata N. Padmanabhan, and Piyush Agrawal. 2008. SixthSense: RFID-based Enterprise Intelligence. In Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys '08). ACM, New York, NY, USA, 253-266. DOI: https://doi.org/10.1145/1378600.1378629

48 Jun Rekimoto, and Nobuyuki Matsushita. "Perceptual surfaces: Towards a human and object sensitive interactive display." In Workshop on Perceptual User Interfaces (PUI'97), pp. 30-32. 1997.

49 Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light, and Alfred Lui. 2015. Designing Connected Products: UX for the Consumer Internet of Things. O'Reilly Media, Newton, MA.

50 Slamtec RPLIDAR A2. https://www.slamtec.com/en/Lidar/A2Spec Last retrieved September 19, 2018

51 Bill Schilit, Norman Adams, and Roy Want. Context-Aware Computing Applications. In Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994. Pages 85–90. IEEE Computer Society.

52 Dan Smith, Ling Ma, and Nick Ryan. 2006. Acoustic environment as an indicator of social and physical context. Personal Ubiquitous Comput. 10, 4 (March 2006), 241-254. DOI=http://dx.doi.org/10.1007/s00779-005-0045-4

53 Timothy J. Smith, Stefan Saroiu, and Alec Wolman. 2009. BlueMonarch: a system for evaluating bluetooth applications in the wild. In Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys '09). ACM, New York, NY, USA, 41-54. DOI: https://doi.org/10.1145/1555816.1555822

54 T. Steel, D. Kuiper and R. Z. Wenkstern, "Context-Aware Virtual Agents in Open Environments," 2010 Sixth International Conference on Autonomic and Autonomous Systems, 2010, pp. 90-96.

55 STMicroelectronics. World's Smallest Time-of-Flight (ToF) Sensor. https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html. Last retrieved September 19, 2018.

56 Joshua Strickon and Joseph Paradiso. 1998. Tracking hands above large interactive surfaces with a low-cost scanning laser rangefinder. In CHI 98 Conference Summary on Human Factors in Computing Systems (CHI '98). ACM, New York, NY, USA, 231-232. DOI: http://dx.doi.org/10.1145/286498.286719

57 M. Vázquez, A. Steinfeld and S. E. Hudson, "Parallel detection of conversational groups of free-standing people and tracking of their lower-body orientation," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 3010-3017.

58 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. DOI: https://doi.org/10.1145/2750858.2804271

59 Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: virtual touch screen in the air using RF signals. SIGCOMM Comput. Commun. Rev. 44, 4 (August 2014), 235-246. DOI: https://doi.org/10.1145/2740070.2626330

60 Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. 1992. The active badge location system. ACM Trans. Inf. Syst. 10, 1 (January 1992), 91-102. DOI=http://dx.doi.org/10.1145/128756.128759

61 Jamie A. Ward, Paul Lukowicz, Gerhard Troster, and Thad E. Starner. 2006. Activity recognition of assembly tasks using body-worn microphones and accelerometers. In IEEE transactions on pattern analysis and machine intelligence, 28, no. 10: 1553-1567.

62 Kathryn Whitenton. "Voice First: The Future of Interaction?" Nielsen Norman Group, November 12, 2017.

63 Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. 2011. Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 1581-1590. DOI: https://doi.org/10.1145/1978942.1979173

64 Andrew Wilson and Hrvoje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In Proceedings of the 23nd annual ACM symposium on User interface soft-ware and technology (UIST '10). ACM, New York, NY, USA, 273-282. DOI: https://doi.org/10.1145/1866029.1866073

65 Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07). ACM, New York, NY, 159-168. DOI: https://doi.org/10.1145/1294211.1294238

66 Robert Xiao, Chris Harrison, and Scott E. Hudson. 2013. WorldKit: rapid and easy creation of ad-hoc inter-active applications on everyday surfaces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 879-888. DOI: https://doi.org/10.1145/2470654.2466113

67 Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 193-196. DOI: https://doi.org/10.1145/2556288.2557017

68 Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott E. Hudson and Chris Harrison. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In Proc. MobileHCI '14. 67-76.

69 Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16). 85-94. DOI: https://doi.org/10.1145/2992154.2992173

70 Robert Xiao, Teng Cao, Ning Guo, Jun Zhuo, Yang Zhang, and Chris Harrison. 2018. LumiWatch: On-Arm Projected Graphics and Touch Input. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 95, 11 pages. DOI: https://doi.org/10.1145/3173574.3173669

71 YDLIDAR X4 360°. https://www.robotshop.com/en/ydlidar-x4-360-laser-scanner.html. Last retrieved September 19, 2018.

72 Hui-Shyong Yeo, Gergely Flamich, Patrick Schrempf, David Harris-Birtill, and Aaron Quigley. 2016. RadarCat: Radar Categorization for Input & Interaction. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). ACM, New York, NY, USA, 833-841. DOI: https://doi.org/10.1145/2984511.2984515

73 Xiangyu Yue, Bichen Wu, Sanjit A. Seshia, Kurt Keutzer, and Alberto L. Sangiovanni-Vincentelli. 2018. A LiDAR Point Cloud Generator: from a Virtual World to Autonomous Driving. In Proceedings of the 2018 ACM on Int. Conf. Multimedia Retrieval (ICMR '18). ACM, New York, NY, 458-464. DOI: https://doi.org/10.1145/3206025.3206080

74 Zulkarnain Zainudin, Sarath Kodagoda and Gamini Dissanayake. "Torso Detection and Tracking using a 2 D Laser Range Finder." (2010). Proceedings of the 2010 Australasian Conference on Robotics and Automation, ACRA 2010.

75 Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17). ACM, New York, NY, USA, 1-14. DOI: https://doi.org/10.1145/3025453.3025842

76 Yang Zhang, Chouchang (Jack) Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++: Room-Scale Interactive and Context-Aware Sensing. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). ACM, New York, NY, USA, Paper 273, 15 pages. DOI: https://doi.org/10.1145/3173574.3173847

77 Junhan Zhou, Yang Zhang, Gierad Laput, and Chris Harrison. 2016. AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16). ACM, New York, NY, USA, 81-86. DOI: https://doi.org/10.1145/2984511.2984568