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

Figure 1. We present SurfaceSight, an approach that enriches Internet-of-Thnings (IoT) experiences with touch, user, and object sensing. We equip existing devices, such as a smart speaker, with LIDAR (A). Next, we perform clustering, tracking and classification (B), which unlocks novel experiences including object sensing (C), touch tracking (D), and person tracking (D). We can also detect which side the user is facing, and an estimate of direction (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 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 a device rests. We can recognize and track a wide array of objects, including finger touches 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.

Author Keywords

Ubiquitous sensing; IoT; Smart Environments;

CCS Concepts

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

CHI '19, May 4-9, 2019, Glasgow,UK

https://doi.org/10.1145/random-alphanumeric-1234

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 alone [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 [60] that might entirely lack on-device controls (e.g., Google Home, Amazon Alexa, Apple HomePod). Regardless of the input modality, the user experience is generally recognized to be cumbersome [47], 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, 48]. 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 commodity 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

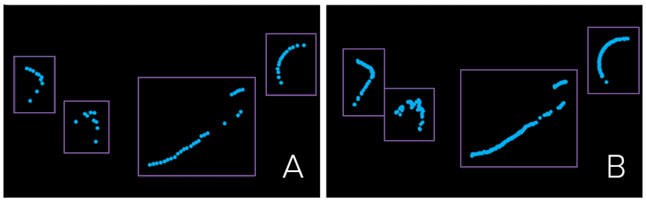

Figure 2. Multi-resolution sampling step. Each LIDAR rotational pass is slightly misaligned. We exploit this property by integrating data from multiple rotational passes. A and B show multi-resolution sampling (B has longer intergration period). Left to right: Mineral spirits, hand, hammer, bowl.

of devices. We illustrate this utility through a set of functional example applications, and quantify the performance of main features in a multi-part user study.

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 more deeply review other systems that have employed LIDAR for input and context sensing, as these are most similar to SurfaceSight in both function and operation.

Ad Hoc Touch Sensing

Research into enabling touch sensing on large, ad hoc surfaces (also referred to as "appropriated" interaction surfaces [18]) goes back at least two decades. By far, the most common approach is to use optical sensors, including infrared emitter-detector arrays [38], infrared cameras [21, 46], depth cameras [61, 63, 66] and thermal imaging [29]. Acoustic methods have also been well explored, using sensors placed at the periphery of a surface [20, 40] or centrally located [65]. Large scale capacitive sensing is also possible with some surface instrumentation (which can be hidden, *e.g.,* with paint), using discrete patches, tomographic imaging [72], and projective capacitive electrode matrices [73].

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 [23], acoustic barcodes [17], RFID tags [34], Bluetooth Low Energy tags and NFCs [14]. Although direct object instrumentation can be fairly robust, they are difficult to scale, and can incur installation and maintenance costs. A complementary approach is to sparsely instrument the environment with cameras [28, 30], radar [69], microphone [50], or through worn sensors [24, 25, 26, 36, 56, 59]. These minimally invasive approaches provide a practical alternative for object and human activity recognition that can power contextually-aware applications.

Person Sensing and Tracking

Many types of systems (from energy efficient buildings [35] to virtual agents [52]) can benefit from knowledge of user

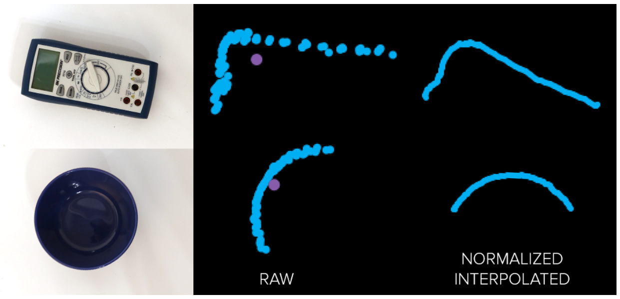

Figure 3. For each cluster, we transform all points into a local coordinate system, rotate, and then resample them for feature extraction. Features include cluster bounding box area, angle between centroid, relative angle between cluster points, and cluster residuals.

presence, occupancy load, and user identification. 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 [19, 58]) or passive (*i.e.*, badge listens for environment signals [11, 42]). Badge-based sensing systems come in other forms, including RFID tags [45], infrared proximity badges [58], microphones [19] and Bluetooth tags [51].

To avoid having to instrument users, researchers have looked at methods including Doppler radar [44], RFID tracking [57] and co-opting WiFi signals [2, 43]. 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 IR proximity sensors [3] and camera-based approaches [6].

Around-Device Interactions

Perhaps most similar to the overall scope of SurfaceSight is the subdomain of Around Device Interaction (ADI). Typically this is for mobile and worn devices, and for capturing touch or gesture input. Several sensing principles have been explored, including acoustics [15, 39], hall-effect sensors [64], IR proximity sensors [5, 22, 27], electric field sensing [74, 31], magnetic field tracking [8, 16], and time-of-flight depth sensing [67]. Across all of these techniques, the overarching goal is to increase input expressivity by leveraging the area around the device as an interaction surface. Our technique complements the rich body of prior work in this space, adding a novel set of interaction modalities and contextual awareness.

LIDAR in Interactive Systems

Originally a portmanteau of light and radar, LIDAR uses the time-of-flight or parallax of laser light to perform rangefinding. 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 [53]. The latter component is an example of a 1D sensor, able to sense distance along a

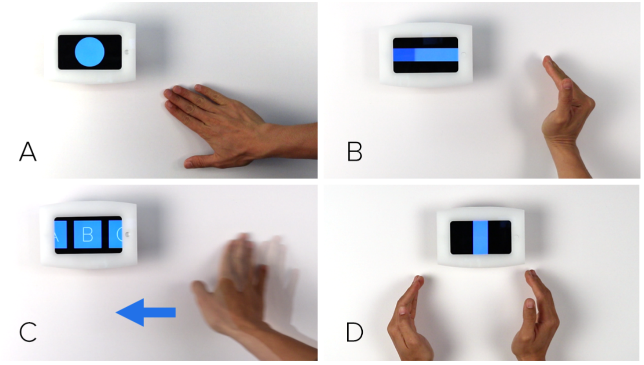

Figure 4. SurfaceSight enables touch and gesture recognition. Here, we show SurfaceSight mounted on a wall. We support a range of modalities, including buttons (A), rotational control (B), carousels (C), and manipulation via two-handed continuous motions (D).

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° [68]). 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 [33]. Solid state LIDAR and wide-angle depth cameras are likely to supersede electromechanical systems in the near future; the interaction techniques we present in this work should be immediately applicable, and likely enhanced with such improvements.

Although popular in many fields of research, LIDAR is surprisingly uncommon in HCI work. It is most commonly seen in human-robot interaction papers, where the robot uses LIDAR data to *e.g.*, track and approach people [32, 55, 71]. 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 [41, 54], first demonstrated in 1997, used spinning LIDAR operating parallel to a surface. Since then, we could only find one other paper, Digital Playgroundz [13], 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.

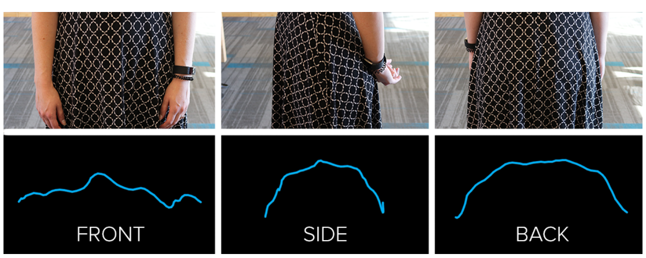

Figure 5. Our system can detect people, including the different sides of their bodies. The signals are quite distinctive, as shown in the bottom row.

IMPLEMENTATION

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

Hardware

For our proof-of-concept system, we build on top of a Slamtech RPLidar A2 [48], 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 cm above the base surface. In a commercial embodiment, we envision the sensor being fully integrated into the base of devices, with a strip of infrared translucent material being used to both hide and protect the sensor.

Multi-Resolution Sampling

The Slamtech RPLidar A2 can sense up to 12 m (15 cm minimum) with its Class 1 (eyesafe), 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°.

Each 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 lag penalty, and also leaves behind "ghost" points if 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 (Figure 2, A and B). First is our "finger" buffer, which integrates five rotations (*i.e.,* 2.4 FPS) for an effective angular resolution of ~0.5°. We found this integration period offers the best balance between robustly capturing small fingers, while still offering an interactive framerate. Our second, "object" buffer, integrates 16 rotational passes (*i.e.,* 0.75 FPS) for an effective angular

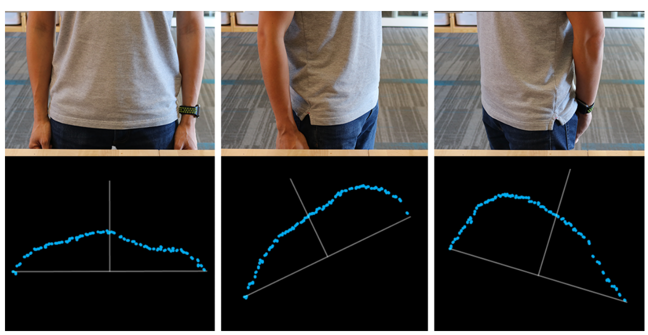
Figure 6. Once we detect that a person is facing *front*, we perform an extra processing step, where we estimate direction. This can be useful as an extra contextual channel to enable devices to discern more explicit user intention.

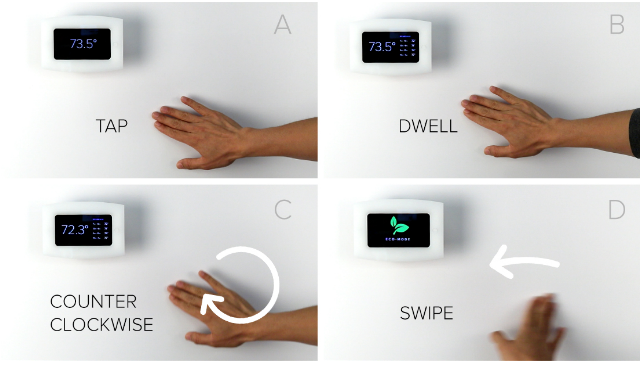

Figure 7. Thermostat demo application. Tapping the wall wakes the device (A), and a dwell activates more details (B). Motion gestures trigger specific commands, such as fined-grained temperature adjustment (C) or moving between different temperature presets (D).

resolution of ~0.2°, which we found strikes a balance between update rate and object contour quality.

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 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.

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. This helps homogenize object contours into a distance-from-sensor and rotation-invariant form. (Figure 3 right).

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.

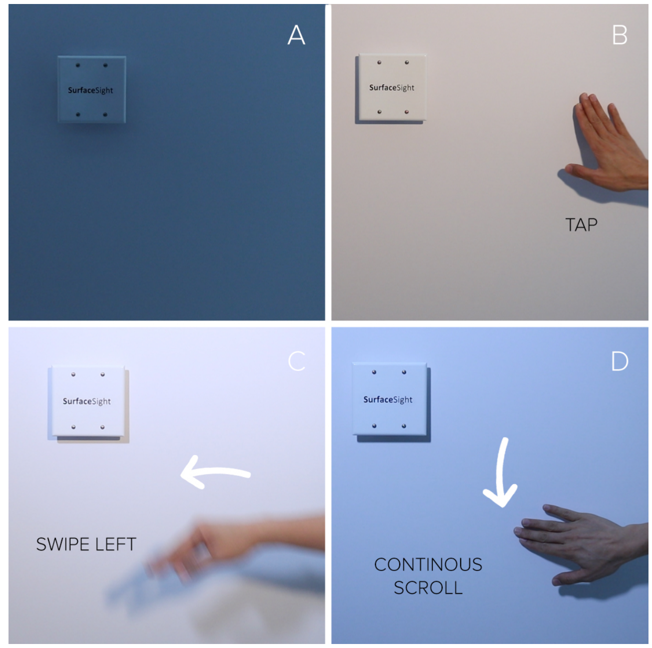

Figure 8. Thermostat demo application. Tapping the wall wakes the device (A), and a dwell activates more details (B). Motion gestures trigger specific commands, such as fined-grained temperature adjustment (C) or moving between different temperature presets (D).

Object Classification & Unknown Object Rejection

Before classification of clusters can occur, a model must be trained on objects of interest. As object contours can be quite different across viewpoints, it is important to expose all sides of an object to SurfaceSight during training. We maintain a database of all previously seen object contours (featurized data), which allows us to compute an incoming contour's nearest neighbor (linear distance function). 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 by using a random forest classifier (in Weka, batch size=100, max depth=unlimited, default parameters).

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. Our tracking pipeline is also responsible for generating *on-down*, *on-move* and *on-lift* events that trigger application-level interactive functions.

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. However, we treat it

Figure 9. Recipe helper demo application. When the recipe is loaded, the system asks the user to retrieve tools (A), and the assistant moves to the next recipe step (B). Contextual questions such as "how many ounces are in this," while referring to a measuring cup (C) are possible. Swipe gestures move between steps (D & E). Finally, when a user is finished with a step (*e.g.,* mortar and pestle lifted), the system can automatically advance.

as a special class of object. Touches to a surface result in conventional interactor events (*e.g.*, on touch down). As noted above, we maintain a movement history of 1.0 seconds for all clusters. In the case of finger inputs, we use this motion vector for stroke gesture recognition. We support six unistroke gestures: *up*, *down*, *left*, *right* swipes, *clockwise, and* counter-clockwise rotations. We modified the $1 recognizer [62] for our unistroke recognition.

In addition to motion gestures, SurfaceSight can also recognize ten *static hand postures* (Figure 11): point, all fingers together, flat palm, fist, wall, corner, stop, 'V', circle, and heart. As these are whole-hand shapes, as opposed to moving figures, we register these contours in our system in the exact same manner as physical objects.

Person Tracking and Direction Estimation

Finally, SurfaceSight can also classify people as another special object class. Human contours are large, move in characteristic trajectories, and are markedly different from inanimate objects (see Figure 5 and 6). We leverage these heuristics for person tracking.

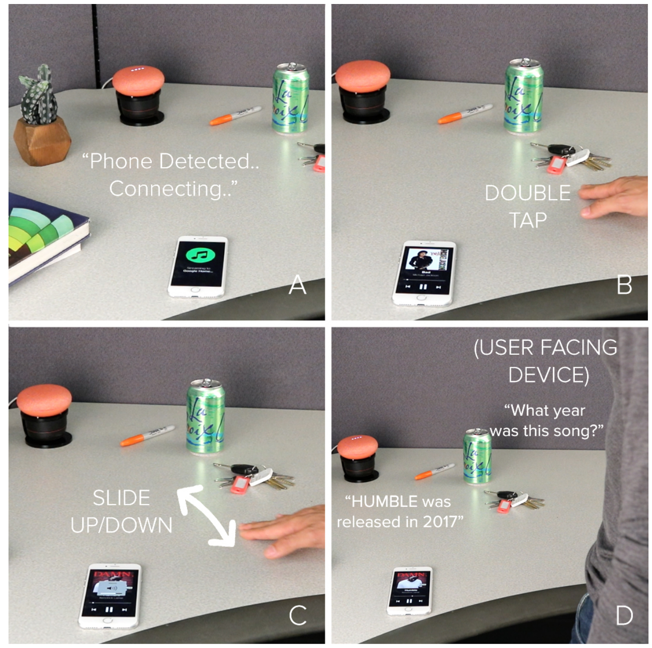

Figure 10. Music player demo application. When the smart speaker detects a person on the table (A), audio streaming is initialized. Gestures such as double tap (B), swipes (C) and continuous scrolling control audio playback. The system can also track the user's body and angle. It listens to user commands (without the wakeword) when it detects that the user is intentionally facing the device.

In addition, we also create three distinct subclasses: person *front, back,* and *side* (Figure 5). If we detect that a person is facing *front*, we perform an extra processing step to estimate which direction they are facing (Figure 6). To compute direction, we project a line between the first and last points of the "human" cluster, and project an orthogonal vector that originates from the midpoint (Figure 6, bottom). Knowing a person's direction can be useful (*e.g.*, accepting voice commands only when a person is nearby and facing the device). From this data, it is also possible to link touch points to a person, as previously shown by Annett and colleagues [3].

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

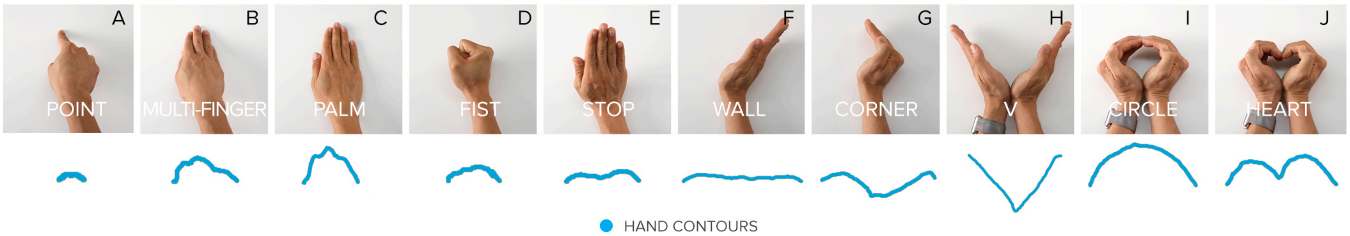

Figure 11. Our ten static hand poses and their corresponding computed features.

the outer perimeter of a work surface, on which we compute a convex hull. Another method we provide is to specify a fixed interactive radius, *e.g.*, 1 meter.

EXAMPLE APPLICATIONS

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

Thermostat

We created a *SurfaceSight*-enhanced thermostat demo that responds to finger touches within a 1 meter radius (Figure 7). Picture frames, a person 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 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.

Lightswitch

For a second wall demo, we created an augmented lightswitch. 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 the wall functions as a dimmer control. Finally, we detect left and right swipes to move between lighting presets, such as incandescent, daylight, evening, and theater.

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 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 and right allows rapid navigation through the recipe steps, including replaying the current step.

Music Player

Finally, we created a music player demo using an instrumented Google Home. This scans for phones resting nearby on the *same* surface (Figure10), 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 spoken input when playing content. In our demo app, the music volume is momentarily halved when a turns to face the Google Home, in anticipation of a spoken command.

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 direction estimation?

For this, we recruited 14 participants (4 female, mean age 29.2), from a public participant pool. The study lasted one hour and paid $10. 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 short-throw projector above the table in order to render automated visual instructions and targets for participants to follow (calibrated to SurfaceSight's coordinate system).

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 randomly positioned crosshair (on a 14 × 6 grid, spaced 15 cm apart, 84 positions total). Users were allowed to use either hand interchangeably, and they were not required to remove accessories, jewelry, or make clothing adjustments. For each trial, we measured the error between crosshair position vs. the touch tracker's position (*i.e.*, cluster centroid). Since our touch tracking is dependent on surface

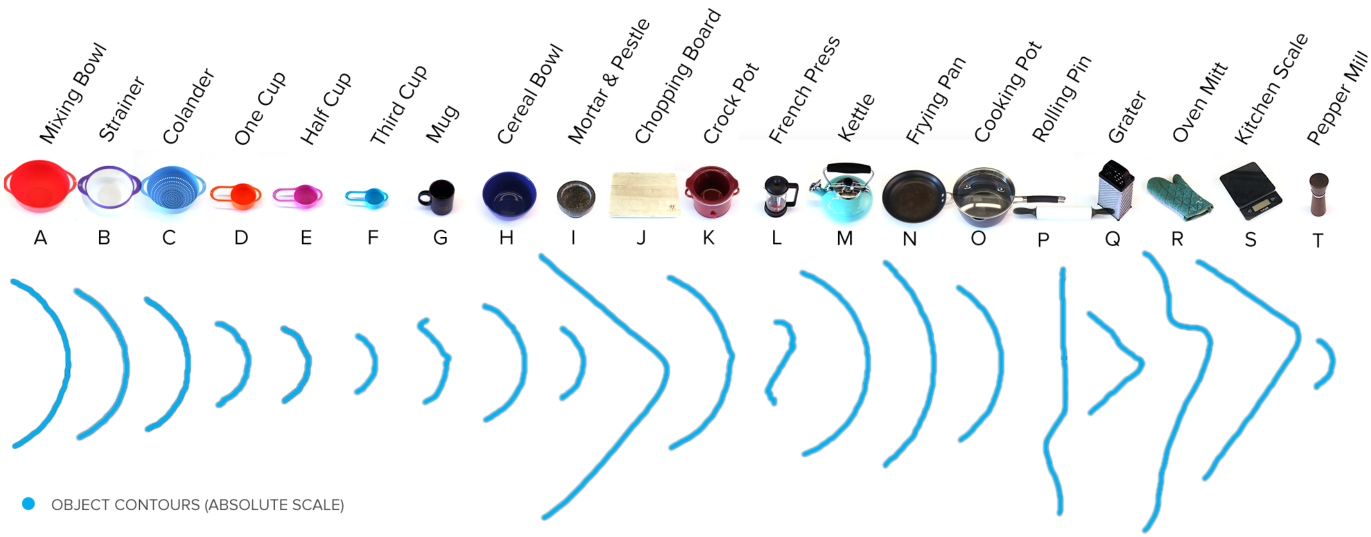

Figure 12. The kitchen object set we used for our SurfaceSight object recognition evaluation. Here, we show the ID, name, and photo of the object, along with a visualization of their normalized-interpolated features.

area, we ran two conditions: touch using a) multiple fingers vs. b) one finger. Across these two conditions, each participant performed 14 × 6 × 2 conditions = 168 trials.

Across 14 users and 2,300 touch trials cumulatively, our system achieved a mean touch accuracy error of ±1.60 cm (SD=0.7cm). We found a linear relationship between touch error and the target's distance from the sensor. There were no significant differences on measurement errors between multiple fingers vs. single finger touch, although false negative errors (*i.e.*, misses) were seen on the single finger condition (*i.e.*, 9.2% missed, SD=5.9%). No touches were missed for the multiple finger condition. The average distance for missed single-finger touches was 1.09 m (SD=0.1m) and 97% of missed touches were 0.8 m away. Overall, these results show the feasibility of touch sensing on SurfaceSight, but caveats still exist. We further discuss the implications of these results in the Limitations section.

	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%	0%	3%	0%	87%	0%	0%	0%	0%	0%	0%	0%	0%	0%
L	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	97%	0%	0%	0%	0%	0%	0%	0%	3%
M	0%	3%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	97%	0%	0%	0%	0%	0%	0%	0%
N	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
O	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
P	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
Q	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	3%	0%	0%	0%	0%	97%	0%	0%	0%
R	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	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 13. Confusion matrix for kitchen objects. Object keys can be found in Figure 12. Overall accuracy across 20 objects is 94.2% (SD=6.9%).

Study #2: Motion Gestures

We also investigated how well SurfaceSight could detect motion gestures. For this task, we defined six directional swipes: a) left, b) right, c) up, d) down, e) clockwise, and f) counterclockwise. Participants performed each gesture twice (in random order), on a 2 × 3 grid (same table). 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 14 users and 1,008 cumulative gesture trials, our system was able to infer dynamic gestures with an accuracy of 97.3% (SD=1.7%). Most gestures achieved an accuracy >98% (most confusion on clockwise vs. down). Most errors occur at far distances, suggesting a slight accuracy decline as gestures are performed further away from the sensor (consistent with findings from the previous study).

	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 14. Confusion matrix for workshop objects. Objects keys are in Figure 15. Accuracy across 18 objects is 91.8% (SD=27.4%).

Figure 15. The workshop object set we used for our SurfaceSight object recognition evaluation. Similar to Figure 12, here, we show the ID, name, and photo of the object, along with a visualization of their normalized-interpolated contours.

Study #3: Static Hand Postures

Beyond motion gestures, we also sought to evaluate how well our system can detect *static* hand postures. For this task, we asked users to perform ten static hand postures, which included single- and two-handed gestures, as depicted in Figure 11. This study was segmented into two parts: a) training and b) testing. In the training phase, users performed all ten gestures in random locations across the table surface, and an experimenter collected data to train a machine learning model (see Implementation section). In the testing phase, a model is trained and gesture inference is performed live. Similar to the previous study, users were asked to perform all ten gestures (random order) on a 2 × 3, and the experimenter captured the system's live prediction. For each grid location, each participant performed 10 gestures = 60 trials. Gesture detection was performed live (*i.e.*, no post-hoc algorithmic change), and the experimenter recorded the system's prediction after each gesture was performed.

Across 14 users and 840 cumulative gesture trials, our system was able to infer static hand gestures with an accuracy of 96.0% (SD=3.01%). We found no significant difference between the gesture detection accuracy vs. location, likely owing to much larger surface area of static hand gestures.

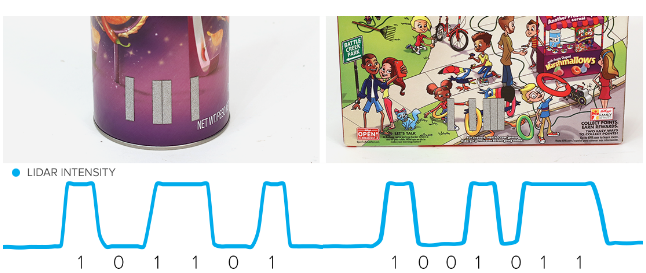

Figure 16. We can tag objects with special retro-reflective material, increasing the set of objects that can be detected by the system. Like barcodes, we can embed data into these tags, as shown here.

Study #4: Body Angle

Next, we sought to evaluate how well our system can detect a person and their relative body angle. For this study, we had seven equally spaced locations around the left, right, and bottom edges of the table. For each location, we display an ellipse (0.5 m diameter) indicating a target. We instruct the participant to move to the target, where we then perform person detection. At the same location, we also project a line on the table surface, and ask participants to align the center of their body towards the projected line (*i.e.*, aligning the center of their hips, torso, nose, and head). We then compare the angular difference between the target line and our predicted angle. We repeat this process three times per location, for a total of 21 trials per user. Similar to the previous studies, predictions were performed live.

Across 14 users and 294 trials, person tracking obtained 100% accuracy. Further, our system predicted body angle accuracy with a mean error of ±3.04° (SD=3.7°). We found no significant difference between the angle prediction vs. location. These results suggest that it is indeed possible to compute the angle of a user's body (albeit when the user is facing the sensor), unlocking novel applications that leverage user directionality as a parameter for device interaction.

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 study, we collected 38 everyday objects, and split them into two functional categories: kitchen and workshop. Similar to the previous study, we segmented this study into training and testing phases. In the training phase, we captured data for each object (different positions, different angles), with ~1000 instances per object. In the testing phase, we trained one model per category, and we performed live prediction. For each trial, a random object, position, and angle was selected, and an experimenter monitored the system's prediction. In all, we captured a total of 1,140 trials (38 objects × 30 random angles and locations).

Across all trials, SurfaceSight garnered and overall object recognition accuracy of 93.1% (SD=6.6%). Recognition accuracy for kitchen objects was 94.2% (SD=6.9%), and 91.8% (SD=27.4%) for workshop objects. Indeed, 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 systems.

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. For example, designating an interactive surface could enable the system to automatically monitor occlusion and provide actionable user feedback (*i.e.*, a clutter detector). Second, we can leverage deep learning-based generative models (*e.g.*, GANs [12, 70]) to "fill-in" measurement gaps between occluded clusters (*e.g.*, when an object is partially blocked), and we plan to explore this approach in future work. 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 only works on level surfaces, and data is inherently planar. 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, and not all objects 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). To mitigate this concern, we implemented custom "tags" attached to objects, allowing them to reflect infrared. We also embedded data into these tags, similar to a low-resolution barcode. In Figure 16, we show how our system decoding an 8-bit binary data from these custom tags.

CONCLUSION

We present SurfaceSight, a new spin on IoT experiences where devices are imbued with rich touch and object sensing. Our system, which incorporates a LIDAR and a full-stack signal processing pipeline, offers an expansive sensing modality immediately above the surface on which a device rests. This capability unlocks expressive input and enhanced contextual awareness, including the detection of objects, finger touches, hand gestures, people tracking, and body angle estimation. Our evaluations reveal the immediate feasibility of our approach, and our example applications illustrate how SurfaceSight can be used to power novel and contextually-aware interactive experiences.

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), Hans-Werner Gellersen (Ed.). Springer-Verlag, London, UK, 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. 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

12. 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.
13. 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
14. 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
15. 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
16. 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
17. 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
18. Chris Harrison. Appropriated Interaction Surfaces. IEEE Computer Magazine, June 2010, 43(6). IEEE, Washington, D.C. 86-89.
19. 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
20. 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
21. 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.
22. 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
23. 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), Clare-Marie Karat, Arnold Lund, Joëlle Coutaz, and John Karat (Eds.). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 550-557. DOI=http://dx.doi.org/10.1145/274644.274718
24. 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
25. Gierad Laput, Karan Ahuja, Mayank Goel, Chris Harrison. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. 2018. In Proceedings of UIST '18.
26. 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
27. 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

28. 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
29. 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
30. 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
31. 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
32. 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
33. 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/
34. 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
35. 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.
36. 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
37. 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
38. 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
39. 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
40. 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.
41. 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.
42. 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
43. 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
44. 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

45. 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
46. Jun Rekimoto, and Nobuyuki Matsushita. "Perceptual surfaces: Towards a human and object sensitive interac-tive display." In Workshop on Perceptual User Interfaces (PUI'97), pp. 30-32. 1997.
47. 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.
48. Slamtec RPLIDAR A2. https://www.slamtec.com/en/Lidar/A2Spec Last retrieved September 19, 2018
49. 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.
50. 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
51. 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
52. T. Steel, D. Kuiper and R. Z. Wenkstern, "Context-Aware Virtual Agents in Open Environments," 2010 Sixth International Conference on Autonomic and Autonomous Systems, Cancun, 2010, pp. 90-96.
53. 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
54. 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
55. 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), Hamburg, 2015, pp. 3010-3017.
56. 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
57. 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
58. 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
59. 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.
60. Kathryn Whitenton. "Voice First: The Future of Interaction?" Nielsen Norman Group, November 12, 2017.
61. 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 software and technology (UIST '10). ACM, New York, NY, USA, 273-282. DOI: https://doi.org/10.1145/1866029.1866073
62. 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, USA, 159-168. DOI: https://doi.org/10.1145/1294211.1294238
63. 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 Com-puting Systems (CHI '13). ACM, New York, NY, USA, 879-888. DOI: https://doi.org/10.1145/2470654.2466113
64. 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

65. 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.

66. 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). ACM, New York, NY, USA, 85-94. DOI: https://doi.org/10.1145/2992154.2992173

67. 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

68. YDLIDAR X4 360°. https://www.robotshop.com/en/ydlidar-x4-360-laser-scanner.html

69. 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

70. 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 International Conference on Multimedia Retrieval (ICMR '18). ACM, New York, NY, USA, 458-464. DOI: https://doi.org/10.1145/3206025.3206080

71. 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.

72. 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

73. 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

74. 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