Extending Mobile Interaction Through Near-Field Visible Light Sensing

Chi Zhang, Joshua Tabor, Jialiang Zhang and Xinyu Zhang
{czhang296,jtabor,jzhang625}@wisc.edu, xyzhang@ece.wisc.edu
Department of Electrical and Computer Engineering
University of Wisconsin-Madison

ABSTRACT

Mobile devices are shrinking their form factors for portability, but user-mobile interaction is becoming increasingly challenging. In this paper, we propose a novel system called Okuli to meet this challenge. Okuli is a compact, low-cost system that can augment a mobile device and extend its interaction workspace to any nearby surface area. Okuli piggybacks on visible light communication modules, and uses a low-power LED and two light sensors to locate user's finger within the workspace. It is built on a light propagation/reflection model that achieves around one-centimeter location precision, with zero run-time training overhead. We have prototyped Okuli as an Android peripheral, with a 3D-printed shroud to host the LED and light sensors. Our experiments demonstrate Okuli's accuracy, stability, energy efficiency, as well as its potential in serving virtual keyboard and trackpad applications.

Categories and Subject Descriptors

H.5.2 [**INFORMATION INTERFACES AND PRESENTATION**]: User Interfaces—*Input devices and strategies*

Keywords

Mobile human computer interaction; visible light sensing; visible light channel modeling; wearable computing

1. INTRODUCTION

Touch has been a de facto mode of input for modern mobile devices like smartphones. Despite its popularity, it carries a number of inherent limitations. Interacting with touchscreens can lead to fingers occluding valuable screen real estate, which is especially cumbersome for real-time applications such as gaming. The problem is being exacerbated given a continued trend for ever-smaller devices. For emerging wearable computing devices, *e.g.*, smart watches, touch input becomes virtually infeasible.

Alternative mobile interaction technologies have been explored to unleash user's finger/hand from the screen. RF-IDraw [1] and PhonePoint [2] use RFID or motion sensors to track hand trajectory, creating opportunities for writing in the air. These approaches can achieve sub-meter scale tracking accuracy, but require instrumenting the hand with additional devices. Vision-based input detection systems [3, 4] achieve finer granularity, by projecting a keyboard or trackpad and extracting user's finger movement with image processing. Yet the need for bulky cameras/projectors compromises portability. UbiK and TapSense [5, 6] piggyback on ambient objects' surface to create a virtual trackpad/keyboard, yet their touching and keystroke identification algorithms rely on tedious run-time training.

Inspired by the emerging visible light communication (VLC) technology [7, 8], we propose Okuli, a novel system that moves interaction off the display and into nearby field around a mobile device. Recent VLC systems have demonstrated capabilities to localize active objects instrumented with light sensors [9, 10]. In contrast, Okuli enables a VLC-capable mobile device to sense passive object, in particular, user's single-finger input. The mobile device uses an LED transmitter to project a light beam towards the finger, and uses photodetectors (PDs) to sense the reflected light intensity, thus localizing the finger position. Okuli affords two immediate interactions for small mobile devices: *(i)* A virtual trackpad that tracks a user's finger, enabling hand-writing based input for smart watches or smartphones with a small display area. *(ii)* A ubiquitous text-entry system that locates user's finger click on a virtual keyboard, *e.g.*, one printed on a flat surface or a piece of paper [5].

In contrast to prior fine-grained input sensing systems that learn location-dependent acoustic/RF signal features [5], Okuli adopts a *model-driven* framework for finger positioning, which is free of run-time training overhead. Okuli achieves this based on two empirical principles originating from physical properties of the visible-light channel. *(i)* Unlike the well-known infrared-based proximity sensing technology that detects "beam cutting" activities [11], Okuli harnesses the non-linear propagation/attenuation effect of light signals, enabling a large input canvas and richer localization information than a binary "presence or absence". *(ii)* Unlike acoustic/RF signals, light propagation bears a more deterministic model, yet this also implies more spatial ambiguity. Okuli thus employs two light sensors to discriminate finger locations that reflect similar light intensity.

Despite its simplicity in principle, Okuli faces a number of key challenges in meeting centimeter-level localization granularity in its anticipated application scenarios. In particular, although light attenuation follows well-defined model in free-space, reflection on irregular-shaped finger, together with irregular LED/PD angular gain patterns, incurs substantial uncertainties. To enable robust finger positioning, Okuli *grooms* the light by reshaping the LED/PD's field-of-view (FoV) with a shroud. It then executes a 3-point initial calibration, based on which it models/predicts the expected light intensity perceived by each PD, when the finger is touching an arbitrary location on the canvas. Okuli's light grooming also enables it to accurately detect finger touching/clicking events.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MobiCom'15, September 7–11, 2015, Paris, France.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-3543-0/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2789168.2790115.

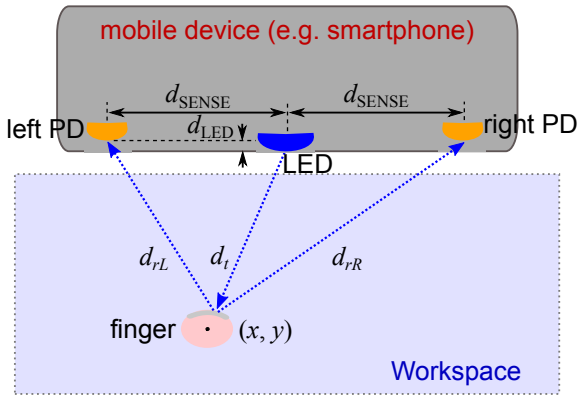

Figure 1: Okuli system architecture.

Uncertainties can also be induced by reflection from irrelevant background objects, and interference from ambient light source. Okuli takes advantage of the flickering capability of VLC transceivers to perform common mode rejection, thus highlighting the finger's impact only. Okuli further leverages flickering to adapt the emitting/sampling rate of LED, thus constraining its energy cost.

We prototype Okuli as a peripheral module to an Android device. The light grooming mechanism is realized via a 3D-printed shroud, which hosts the LED/PDs. We use near-infrared (NIR) LED and PDs in the prototype, whose wavelength is very close to that of visible light, and has identical channel characteristics. The driver circuit for light projection/sensing interfaces with a tablet running virtual keyboard/trackpad applications. All signal processing and localization algorithms run inside the tablet. We expect a product-level implementation can easily embed the LED/PDs on the side of a mobile device, and use part of its body as shroud.

Our experiments on this prototype demonstrate Okuli's great potential in realizing near-field mobile interaction. On a 9cm×7cm region, equivalent to a small trackpad, it can detect and localize random finger positions with a median error of 0.7cm, and 90-percentile error of 1.43cm. It shows consistent precision over different surfaces. Given LEDs/PDs with wider FoV and higher power/sensitivity, Okuli's working region can be further expanded.

We have also prototyped the aforementioned keyboard and trackpad applications on top of Okuli. Our experiments show that handwriting tracked by Okuli achieves a character recognition rate of 90.6%, comparable to that from a tablet. In addition, Okuli realizes a small keyboard within its workspace, with keystroke recognition rate of 90.7%. A field test with multiple users demonstrates that Okuli maintains consistent performance in the presence of different fingers and usage behaviors.

The key contributions of Okuli can be summarized as follows.

(i) Proposing a model-driven framework, facilitated by a light-grooming mechanism, to realize fine-grained finger positioning using visible light signals. The model also enables us to examine salient system properties including spatial ambiguity and dimension scalability.

(ii) Designing mechanisms to eliminate the interference from background reflection and ambient light, and limiting the power consumption of a passive light sensing system.

(iii) Implementing and validating the entire system as a smartphone peripheral with a 3D-printed shroud. Performing case studies of Okuli in trackpad/keyboard applications.

2. Okuli OVERVIEW

System Architecture. Okuli is a low cost passive finger localization system based on visible light sensing. As shown in Figure 1,

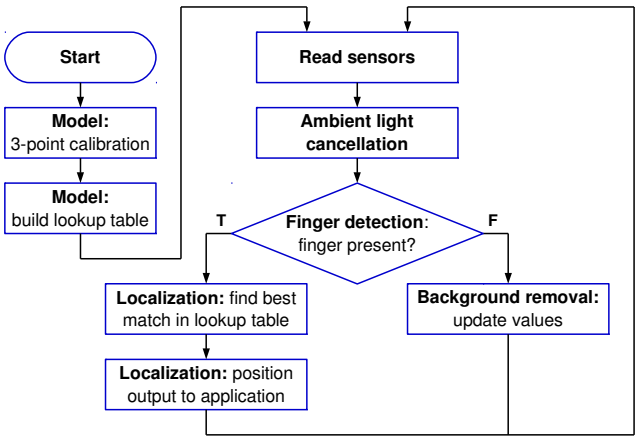

Figure 2: Simplified workflow of Okuli.

Okuli consists of 1 light source and 2 light sensors. The light source is an LED, and sensors are photodetectors (PDs) whose output increases linearly with input light intensity. The LED is centered between the 2 PDs which are typically separated well below 10cm so as to be fitted into the edge of a mobile device. We refer to the straight line between the two PDs as Okuli's *front face*.

Okuli uses the region in front of this line as the *workspace*. The workspace's coordinate system uses the center-point between two PDs as the origin, and its x-axis is parallel to Okuli's front face. Finger position within the workspace is denoted with (x, y). The LED can be placed slightly in front of the front face if desired. In the rest of this paper, we denote the distance between the origin and each PD as d_{SENSE}, and that between the origin and the LED as d_{LED}. The distances between the finger and the LED, left and right PD are denoted by d_t, d_{rL} and d_{rR}, respectively.

Usage conditions. Since Okuli uses a light propagation/reflection model for finger localization, its accuracy highly relies on the accuracy of the modeling parameters. However, Okuli will perform consistently as long as the following conditions are satisfied: *(i)* The reflectivity of user's finger does not change significantly over time. *(ii)* User maintains a relatively consistent usage habit (*e.g.*, same finger), and the hand/wrist does not touch the working surface, just like when using a touchscreen. *(iii)* The workspace stays on the same surface with a consistent reflectivity.

From our user study, we found that these conditions are naturally met in Okuli's targeted application scenarios, and users naturally maintain the consistency needed. When the usage conditions are violated, Okuli requires the user to input 3 anchoring points to recalibrate the modeling parameters (Section 4.3).

Design goals and challenges. Okuli needs to achieve the following goals in order to facilitate mobile interaction:

(i) Fine-grained, model based finger localization: Okuli aims for model-driven localization with around one-centimeter accuracy (around the size of a key on a keyboard), and without a fingerprinting-like training procedure. It needs to account for all factors that can affect the light intensity reflected by the finger into the PDs. These include not only finger-to-PD distance, but also the LED/PDs' angular response, finger reflectivity, *etc.*

(ii) Robustness: Okuli's model should be stable over time. In addition, since Okuli uses visible light to locate the finger, two kinds of visible light sources may interfere our system: (a) Ambient light such as fluorescent lamp and sunlight; (b) Reflection from background object other than the finger. The first type of interference could be easily canceled due to its stability, but the second type imposes challenges since it is "coherent" with the finger reflection and may be dynamic.

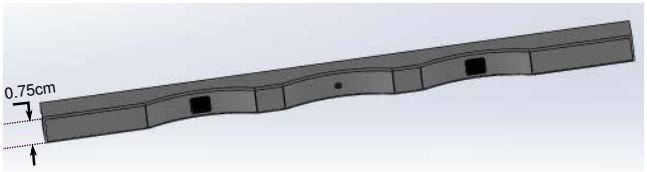

Figure 3: Concept-art of a product version of Okuli, to be fit into the edge of a small mobile device with a thickness around 7.5mm.

(iii) Power consumption and form factor: Okuli should have a relatively low power consumption to be equipped on battery-powered devices. In addition, Okuli should have a small dimension so as to be integrated to small mobile devices such as smartphones and smart watches.

System workflow. To achieve the above goals, Okuli incorporates the following four signal processing modules on top of its hardware substrate: visible light channel modeling and calibration, ambient light cancellation, finger detection and background removal, and finger localization. Figure 2 illustrates a simplified workflow of Okuli.

The *modeling/calibration* module (Section 4) requires the user to conduct a 3-point initial calibration procedure. The calibration sets user specific parameters in the light propagation/reflection model. The model then builds a lookup table that predicts sensor output when the finger is located at an arbitrary point within the workspace.

The *ambient light cancellation* module (Section 5.1) suppresses the interference caused by other light sources. It works by turning the LED on and off and then calculating the difference of sensor outputs, thus eliminating the components contributed by external light sources.

The *finger detection and background removal* module (Section 5.2) first determines whether user's finger is present. Upon detecting the finger, it invokes the finger localization module; Otherwise, it opportunistically updates the estimation of dynamic background values, which result from reflections upon nearby objects such as user's body.

Finally, at run-time, the *finger localization* module (Section 5.3) compares the PDs' outputs with those in the lookup table and determines the most likely location of the finger, *i.e.*, the position that has a predicted sensor output that best matches the measured one.

3. DISCIPLINING THE LIGHT SENSING

3.1 Field-of-View (FOV) Consideration for Visible Light Sensing

Okuli uses a visible light source and dual PDs to locate a finger within its working space. If the light sensing is not carefully controlled, Okuli can become very susceptible to background reflections and ambient light interference. The former can be attributed to reflections from user's hand, arm or other objects close to Okuli. The latter can saturate Okuli's PDs and, if the ambient light is dynamic, can lead to false detections.

To alleviate these problems, Okuli should ideally use PDs with a very wide horizontal FOV and a very narrow vertical FOV. The wide horizontal FOV creates a wide workspace, and the narrow vertical FOV helps reduce background and ambient light interference by limiting the amount of objects capable of reflecting light into Okuli's PDs. By constraining the FOV of the PDs, Okuli only views a small slice of the user's finger, plus a limited amount of interference.

FOV can be shaped through custom-built PDs or lenses, which can be made extremely small and fit comfortably on the side of a smartphone or smart watch. Figure 3 shows an example of how Okuli could look in an actual product-level implementation.

3.2 A 3D Printed Shroud for Light Grooming

Since there exist no PDs with the required FOV characteristics on the market, a shroud is 3D printed to hold the PDs and discipline their FOV. Figure 4 shows a CAD drawing of the shroud. We have printed the shroud on a Dimension Elite 3D printer using ABS thermoplastic painted in black.

The PDs' vertical FOV is limited by the slits on the left and right side of the shroud. The light source, an LED, sits in the center of the shroud and is blocked from the PDs to prevent light-of-sight leakage. The shroud is painted black to prevent complicated reflections and to reduce the impact of ambient light diffusion/interference. It allows the PDs to be mounted in different positions. Such extra flexibility make the prototype larger than what would be necessary in a product version.

Figure 5 and Figure 6 show the horizontal/vertical angular responses of a PD, with and without the shroud. We observe that the shroud grooms the vertical FOV into a super-narrow, non-divergent beam, while forcing the horizontal FOV within 90°. Thus, it achieves the same effect as an ideal PD for Okuli.

We also observe that the shroud helps alleviate the impact of background reflection from the user's hand. As shown in Figure 7, without the shroud, the received light intensity shows large variation with the same finger position but different hand postures. The shroud significantly reduces such variation and enables stable finger-localization.

4. MODELING NEAR-FIELD LIGHT REFLECTION

To achieve localization, Okuli uses a model to estimate the intensity of reflected light when a finger is placed at a specific coordinate within the workspace.

Light emitted by the LED is subject to 4 processes before reaching the PD: emission, propagation, reflection and reception, which correspond to 4 attrition factors: LED's angular response; pathloss between LED, finger and PDs; reflectivity of finger; and finally, PD's angular response. The final light intensity sensed by the PD is a multiplication of the 4 attribution factors. We model the attrition factors in three parts: a *finger response estimator*, a *free-space path loss estimator* and *parameter calibration*, respectively.

For a given finger position (x, y), the *finger response estimator* (Section 4.1) estimates the percentage of light reflected by the finger. The LED and PD are at different angles for different (x, y) coordinates, resulting in different amounts of light being reflected and received.

The *free-space loss estimator* (Section 4.2) estimates the light's attenuation as it traverses the illuminating path and reflecting path.

Finally, the *parameter calibration* mechanism (Section 4.3) acquires hardware and user-specific attrition factors. Through a one-time measurement, it obtains normalized angular responses of the LED and PDs, which vary from advertised ones, and can be changed radically after light grooming. In addition, a constant $C_X (X \in \{L, R\})$ for left/right PD is used to capture other parameters, *e.g.*, the reflectivity of the finger and the coefficients for the real gain of the LED and PD. We emphasize that the calibration procedure only needs to be conducted at factory-production stage, and when Okuli serves a new user or new workspace.

4.1 Finger Response Estimator

Okuli uses a *finger response estimator* to model the amount of light reflected from a user's finger. There are two major factors that

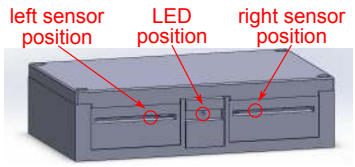

Figure 4: CAD model of the front view of our Okuli prototype. Note only the PDs' slot width (2mm) and depth (3mm) determine the dimension of a product-level implementation.

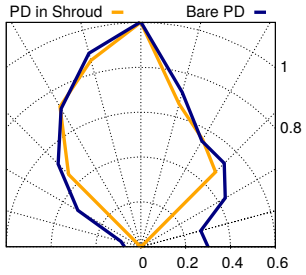

Figure 5: Horizontal angular response of bare PD and PD in shroud.

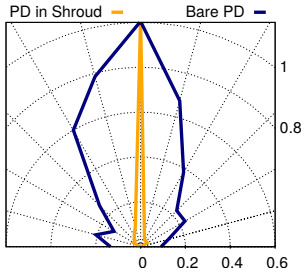

Figure 6: Vertical angular response of bare PD and PD in shroud.

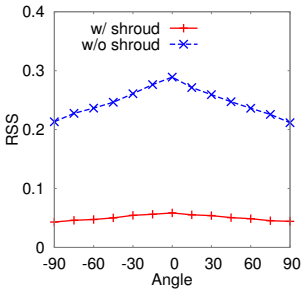

Figure 7: RSS of one PD with user's hand posture in different angles.

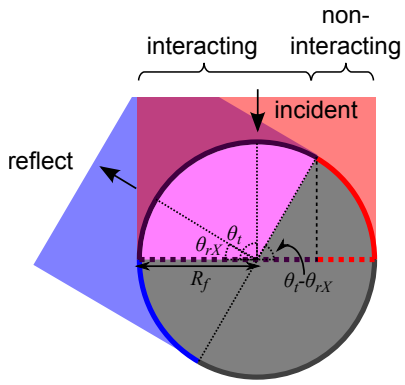

Figure 8: Modeling the *reflection coefficient* of the finger. R_f is the approximate radius of the finger.

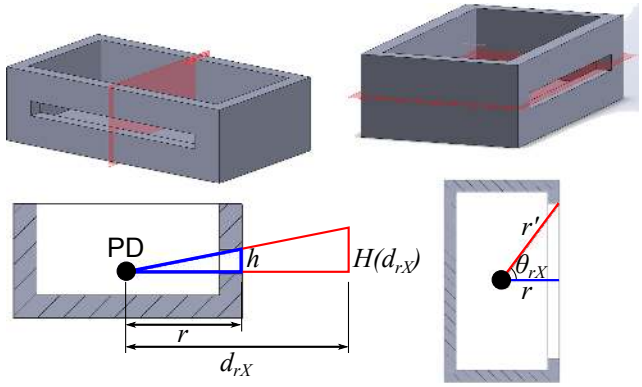

Figure 9: Illustration describing Equation (5). The top of shroud is removed for illustration purpose. To generate the cross-sections at the bottom, the shroud is cut by the red plane.

shape the finger response. First, the finger receives and reflects light in different angles at different locations, which can lead to different overall reflectivity. Second, the distance between user's finger and Okuli's front face determines how much of the finger can be seen by Okuli, and hence how much light is reflected. Accordingly, the finger response estimator comprises two essential coefficients: the *reflection coefficient* and the *visible height coefficient*.

4.1.1 Modeling the Reflection Coefficient

The reflection coefficient models how the finger "bends" incoming light. It can be represented as:

$$C_r \cdot \frac{\textit{Light Reflected towards PD}}{\textit{Light Reflected in All Directions}} \tag{1}$$

where C_r is a user-related constant that represents skin's reflectivity and is calculated during calibration (Section 4.3). The remainder fraction term falls in the range $[0, 1]$, representing the percentage of the light arriving at the finger that is reflected on to the PD.

As illustrated in Figure 8, this fraction term is determined by incident and reflecting angles. It can be modeled using a function $R_X(\theta_t, \theta_{\mathrm{rX}})$, where θ_t denotes the angle between the incidental and reflecting light directions, and $\theta_{\mathrm{rX}}(X \in \{L, R\})$ the angle between the reflection direction to left/right PD and the horizontal plane (parallel to Okuli's front face). For each finger location (x, y), θ_t can be calculated by simple geometry based on Okuli's setup (Figure 1, where $x_{\mathrm{LED}} = 0$, $y_{\mathrm{LED}} = d_{\mathrm{LED}}$, $x_{\mathrm{PD_X}} = \pm d_{\mathrm{SENSE}}$ and $y_{\mathrm{PD_X}} = 0$):

$$\theta_t = \arctan\left(\frac{y - y_{\mathrm{LED}}}{x - x_{\mathrm{LED}}}\right) \tag{2}$$

Similarly, θ_{rX} can be calculated as:

$$\theta_{\mathrm{rX}} = \arctan\left(\frac{y - y_{\mathrm{PD_X}}}{x - x_{\mathrm{PD_X}}}\right) \tag{3}$$

From Figure 8, $R_X(\theta_t, \theta_{\mathrm{rX}})$ equals the length of the projection of the incident and reflecting beams (red and blue areas respectively), divided by the total width of the beam received from the LED, which is $2R_f$. Consequently, we have:

$$\begin{aligned} R_X(\theta_t, \theta_{\mathrm{rX}}) &= \frac{R_f + R_f \cdot \cos(\theta_t - \theta_{\mathrm{rX}})}{2R_f} \\ &= \frac{1}{2}\left[1 + \cos(\theta_t - \theta_{\mathrm{rX}})\right] \end{aligned} \tag{4}$$

which only depends on the angle between the incident and reflected beam. Intuitively, the coefficient achieves its maximum when $\theta_t - \theta_{\mathrm{rX}} = 0^\circ$ as both the LED and the PD see the whole vertical cross section of the finger, and goes to zero when $\theta_t - \theta_{\mathrm{rX}} = 180^\circ$ as the light from the LED is totally blocked by the finger.

4.1.2 Visible Height Coefficient

The *visible height coefficient* estimates the total vertical height visible to both the LED and the PD. Since the LED's vertical FOV is much larger than the one of the shroud and thus relatively unrestricted, the PD's FOV in the shroud is the sole factor in determining this coefficient, which compensates for the fact that more of the user's finger is visible the farther it is away from the PD, so more light can be reflected from the finger towards the PD.

Figure 9 illustrates the derivation of the visible height coefficient. In the bottom left of Figure 9, r represents the distance between the PD and the opening of the shroud, and h is the height of the opening. $d_{\mathrm{rX}}(X = \{L, R\})$ denote the distance from the finger to left and right PD, respectively. The inner blue triangle and the outer red triangle are similar, so a simple proportion can model the *visible height coefficient* in terms of d_{rX}, r and h:

$$H(d_{\mathrm{rX}}) = \frac{d_{\mathrm{rX}} \cdot r}{h} \tag{5}$$

Note that r changes with the angle θ_{rX}: as shown in the bottom right of Figure 9, for a finger position near the edge of the PD's FOV, corresponding $r' = \frac{r}{\cos\theta_{\mathrm{rX}}}$, which changes Equation (5) to be:

$$H(d_{\mathrm{rX}}) = \frac{d_{\mathrm{rX}} \cdot \frac{r}{\cos\theta_{\mathrm{rX}}}}{h} \tag{6}$$

However, since this change only affects the overall angular response of the PD, it will also be accounted for during calibration. So the simpler Equation (5) suffices, *i.e.*, r and h can be considered as constant parameters.

4.1.3 Assumptions in the Finger Response Estimator

We have made two assumptions in the finger response estimator to simply the model. First, the reflection coefficient is accurate only if the finger is relatively round, as the reflection coefficient assumes a circular cross-section. Second, it is assumed that the finger is a good diffuser of light.

To verify the first approximation, we place a finger vertically in front of Okuli, and measure the RSS as the finger rotates. If the finger were a perfectly round, uniform reflector, the RSS would change very little with the rotation. Figure 7 shows how little RSS changes as the user rotates finger (and hand) within $\pm 90^\circ$. In practical usage scenarios, a user will rarely rotate finger beyond this range. So the round-finger assumption can be used without a considerable loss in accuracy.

In addition, the *visible height coefficient* assumes the finger is a diffusive reflector, and hence the visible height grows with distance. If the finger instead creates specular reflection, none of its surface area above the light source would reflect light into the PD, resulting in a constant height. Figure 10 shows the finger part lit by a camera in a dark room. The finger is evenly colored, which testifies to the diffusion effect. Assuming fingers are perfect diffusers adds some small inaccuracies, but it also greatly simplifies the model.

4.2 Free-Space Path Loss Estimator

The second component of Okuli's modeling framework is the *free-space loss estimator*, which estimates the light intensity attenuation due to traveling through free-space. The free-space loss along the illuminating path follows the inverse-square law for visible light propagation [12]. We denote this loss as g_t and the distance between the LED and user's finger as d_t, then we have $g_t = 1/d_t^2$.

On the other hand, we denote the reflection path from the finger to the PD as g_{dX}, $X = L$ and R for left and right PD, respectively. This path involves a more sophisticated model that needs to account for the increasing amount of finger surface area visible to the PD as the finger moves farther. Taking into account this phenomenon along with the finger response estimator and the inverse-square law, we can derive the reflection pathloss as:

$$\begin{aligned} g_{\mathrm{dX}} &= \frac{R(\theta_t, \theta_{\mathrm{rX}}) \cdot H(d_{\mathrm{rX}})}{d_{\mathrm{rX}}^2} \\ &= \frac{\frac{1}{2}[1 + \cos(\theta_t - \theta_{\mathrm{rX}})] \cdot C_r \cdot \frac{d_{\mathrm{rX}} \cdot r}{h}}{d_{\mathrm{rX}}^2} \\ &= \frac{[1 + \cos(\theta_t - \theta_{\mathrm{rX}})] \cdot C_r \cdot r}{2 d_{\mathrm{rX}} \cdot h} \end{aligned} \tag{7}$$

From Equation (7), it is clear that *the reflection pathloss follows an inverse-linear law instead of the common inverse-square law, due to change in the finger's visible height.*

4.3 Model Calibration

Finally, to conduct the model calibration, we first merge hardware-related parameters (*e.g.*, the output power of LED), user-related parameters (*e.g.*, variation in finger reflectivity caused by different skin color), and workspace-related parameters (*e.g.*, surface reflectivity), together into a constant, which we denote as $C_X (X \in \{L, R\})$ for each PD. Also, we denote the signal output of each PD as $\mathrm{RSS}_X (X \in \{L, R\})$, beam shape (angular response) of the LED as $S_{\mathrm{LED}}(\theta_t)$, and angular response of the sensor as $S_{\mathrm{PD_X}}(\theta_{\mathrm{tX}}) (X \in \{L, R\})$. Following our discussion in the beginning of Section 4, the final model can be expressed as:

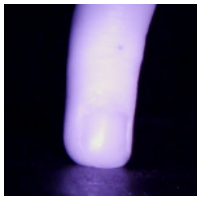

Figure 10: Diffusion of light source on finger as viewed by a camera. We have adjusted the exposure to ensure that the camera is not saturated.

$$\mathrm{RSS_L} = C_L \cdot g_t \cdot g_{\mathrm{dL}} \cdot S_{\mathrm{PD_L}}(\theta_{\mathrm{rL}}) \cdot S_{\mathrm{LED}}(\theta_t) \tag{8}$$

$$\mathrm{RSS_R} = C_R \cdot g_t \cdot g_{\mathrm{dR}} \cdot S_{\mathrm{PD_R}}(\theta_{\mathrm{rR}}) \cdot S_{\mathrm{LED}}(\theta_t) \tag{9}$$

Equations (8) and (9) involve parameters that need calibration before being used to estimate the expected RSS's at different locations. The calibration follows two stages.

The first stage consists of experimentally determining the angular responses of both the LED (*i.e.*, $S_{\mathrm{LED}}(\theta_t)$) and the PD (*i.e.*, $S_{\mathrm{PD_X}}(\theta_{\mathrm{rX}}), X \in \{L, R\}$) after they are mounted in the shroud. To obtain the LED response, we simply point a PD towards it, rotate the PD along a half-circle while measuring the RSS at different angles. To obtain the PD response, we use LED to do the measurement in a reciprocal way. This is a one-time factory calibration procedure, and is necessary for two reasons: the manufacturer-reported angular response does not match the actual device very well, which can be clearly seen from the irregular response curve of the PD before applying the shroud (Figure 5); also, the shroud significantly changes the responses of both the LED and the PD.

Given the angular responses, we proceed to the second stage. Since we model the reflection light intensity as the transmitted light intensity $S_{\mathrm{LED}}(\theta_t)$ multiplied by a chain of attrition factors, we can extract all the user-related and workspace-related constants and multiply them into a single factor, which is represented in the aforementioned constant C_X. The calibration fits 3 sample inputs from the user in order to obtain the parameter C_X. While the user inputs samples for 3 known anchoring points within the workspace, RSS values are taken from each PD, and the constant C_X is calculated for each point by plugging in the measured RSS values into Equations (8) and (9) to solve for C_X. To avoid biasing the model to recognize one point better than the others, the averaged C_X value across the 3 anchoring points is used. If the user misplaces her finger or otherwise completes calibration incorrectly, a sanity check of the calibration points is done by taking standard deviation of the individual C_X values. If the standard deviation is higher than an empirical threshold, then the user is asked to reinitiate the calibration.

The resulting C_X accounts for the shape and reflectivity of each user's finger, the gain and aging of the LED and PD, and the reflectivity of the surface Okuli is placed on. Therefore, Okuli needs to be recalibrated when placed on a new surface, or used by a different person.

5. FINGER POSITIONING

In this section, we introduce how Okuli incorporates the above model to establish a practical finger localization framework. Specifi-

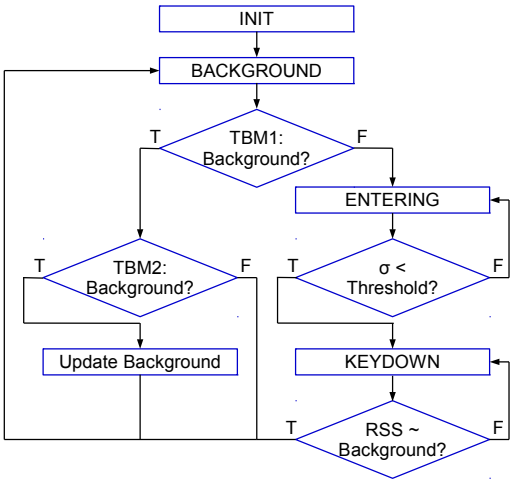

Figure 11: Flow chart for dynamic background removal.

cally, we first describe how Okuli suppresses ambient interferences to obtain the RSS of desired signals, *i.e.*, signals solely due to finger reflection and pertinent to our model. Then, we discuss the localization mechanism, analyze its error tolerance and scalability in larger workspace. Finally, we briefly analyze Okuli's energy efficiency.

5.1 Canceling Ambient Light Interference

In practical usage scenarios, Okuli may be subject to a variety of ambient interferences, which can be broadly classified into ambient light interference and irrelevant background reflection. This section tackles the former.

Okuli uses a simple cancellation mechanism to suppress interference from ambient light sources such as sunlight and indoor fluorescent/LED lights. It modulates the LED's light emission using periodic On-Off-Keying (OOK), with a switching rate of around 30Hz — much higher than the typical variation rate of ambient light sources. During the ON period, Okuli's PD takes one RSS sample that contains both the desired signal and ambient light interference, and subtracts the sample from previous OFF period, which only contains the ambient light intensity. Thus, light intensity contributed by ambient light is removed from the measured value.

We note that the ambient light cancellation mechanism may not work under extreme conditions, such as direct sunlight, where ambient light can easily saturate the ADC. An analog subtracter built with multiplexers and operational amplifiers, which are synchronized with the modulation, might overcome this challenge. We leave the exploration of such schemes for future work.

5.2 Dynamic Background Removal

In addition to ambient light interference, it is also possible that there exist interferences "coherent" to Okuli's modulated light, interferences that are emitted during the on period and dark during the off period. Any secondary sources of reflection other than the target finger can cause such interference. Examples include reflection from user's body parts/clothes, the working surface, people walking by, and even self-interference due to insufficient isolation between the LED and the PD. We colloquially refer to these interferences as *background*.

To remove the background interference, we can assume that the background does not change very quickly (within a few sampling periods of the PD), and Okuli can keep track of the background readings when the finger is absent from the workspace. Then, Okuli can subtract the latest measurement of background interference from the measured RSS values. We call this approach *dynamic background removal* (DBR). The approach requires two essential mechanisms: finger presence detection and background estimation.

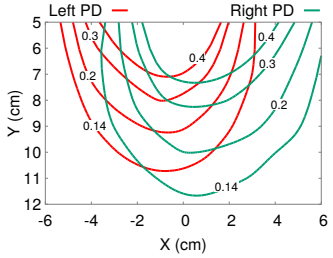

Figure 12: Contour plot of the RSS spatial distribution of the left and right PD at different locations.

Finger presence detection. To realize DBR, we need to identify the presence of finger and classify each RSS reading into 2 cases: foreground and background. We use the Temporal Background Modeling (TBM) in [13] for the classification. The TBM was originally borrowed from computer vision and image processing, to help identify and remove slow-changing background from signal. It uses a Gaussian kernel and a set of parameters to determine how likely a certain sample is the background. For image processing, either a decrease or an increase of values can be considered as a foreground event. However, for Okuli, only an increasing RSS can indicate a finger-presence event in the foreground. Thus, we added an exponential term to the Gaussian kernel in TBM to penalize classifying reducing RSS as foreground.

We design the DBR in conjunction with *finger-touch detection*, as shown in the work flow in Figure 11. It has 4 states: `INIT` (initialization), `BACKGROUND`, `ENTERING` and `KEYDOWN`. During initialization, a circular buffer is filled with a certain amount of samples. In the background state, after each sample update, `TBM1` which favors background assertions is run first. If its output indicates a likely background presence, a second `TBM2`, which is stricter and favors non-background assertions, is executed. If the outcome still indicates high likelihood of background presence, then the background value is updated.

Meanwhile, if `TBM1` indicates a non-background situation, the state will be switched to `ENTERING`. During this state, the finger is supposed to be on its way to touch the working surface, which generates a large standard deviation relative to the mean value. Thereafter, touch occurs, moving the system into the `KEYDOWN` state.

Background estimation. Okuli opportunistically switches back to the `BACKGROUND` state, and updates the background value, whenever the finger leaves the surface. We expect that a finger staying on the working surface for an extended time will be rare. Thus, Okuli should be able to update the background value in a timely way.

5.3 Spatial Ambiguity and System Dimensioning

Eliminating spatial ambiguity. The basic principle of Okuli's localization mechanism is to map a given light intensity measurement into finger location, based on the aforementioned light propagation/reflection model. Note that a single pair of LED/PD is insufficient for location discrimination as there exist many finger spots with the same reflecting light intensity. Okuli eliminates such spatial ambiguity with two PDs.

Figure 12 shows a contour plot of spatial distribution of the RSS, for left and right PD (separated by 2.5 cm), respectively, under an example setup. The contours are generated using the aforementioned model, with parameters calibrated in our real Okuli prototype. We can see that any curve from the left PD has only one intersection with any curve from the right PD. Therefore, *any location* (x, y) *can be mapped to a unique pair of RSS values, and hence, two PDs*

plus one LED are sufficient to discriminate locations on Okuli's 2-D workspace. In general, suppose the PDs' angular responses are homogeneous and close to half-circle, it can be proven using simple geometry that this one-to-one mapping strictly holds [14]. Practical angular responses may deviate from half-circle. However, we conjecture that as long as they are convex and homogeneous, spatial ambiguity can be eliminated given two appropriately separated PDs. We leave the proof for the conjecture as future work.

Location matching. To map the measured RSS pair to location, we note that the RSS pair (RSS_L, RSS_R) provided by model (Equations (8) and (9)) does not have a closed form, and hence it is impossible to directly inverse the RSS to location. So we use a *location matching* mechanism instead. We first use the model to compute a look-up table that maps every location spot on 1mm×1mm grid into an RSS pair. At run-time, Okuli can look for the RSS pair that matches closely with its measured one, and then reverse it to a location spot. The error metric for matching can be either Euclidean distance or Manhattan distance.

The foregoing discussion on spatial ambiguity assumes Okuli's model is perfect. In practice, Okuli's location precision depends on the separation between the target location spots, and also the separation between the LED and PD. We now analyze how such dimensioning parameters affect Okuli's performance.

Error tolerance for a given separation between spots. For a given location (x, y), based on Equations (8) and (9), we can obtain the expected RSS of left and right PD, denoted by L_{xy} and R_{xy}. Since all factors in Equations (8) and (9) are multiplicative, we assume the errors in modeling L_{xy} and R_{xy} are multiplicative Gaussian variables with zero mean and variance σ^2. We denote the two variables as N_L and N_R, respectively. Further, we denote L'_{xy} and R'_{xy} as the actual RSS of left and right PD with modeling error for a given finger location (x, y). Then we have

$$L'_{xy} = (1 + N_L)L_{xy} \tag{10}$$

$$R'_{xy} = (1 + N_R)R_{xy} \tag{11}$$

We use normalized Euclidean distance as the modeling error metric, denoted as RSS_e:

$$\begin{aligned} RSS_e &= \sqrt{\frac{\left(L'_{xy} - L_{xy}\right)^2}{L_{xy}^2} + \frac{\left(R'_{xy} - R_{xy}\right)^2}{R_{xy}^2}} \\ &= \sqrt{N_L^2 + N_R^2} \end{aligned} \tag{12}$$

If we further assume N_L and N_R are independent, then RSS_e follows Rayleigh distribution with variance σ^2. The kth-percentile p of Rayleigh distribution is known to be [15]:

$$p = \sigma \sqrt{\ln\left[\left(1 - \frac{k}{100}\right)^{-2}\right]} \tag{13}$$

To distinguish two nearby locations (x_1, y_1) and (x_2, y_2) with probability larger than $k\%$, the percentile of modeling error of Okuli should be smaller than the normalized Euclidean distance of RSS between the two locations:

$$p < \sqrt{\frac{\left(L_{x_2y_2} - L_{x_1y_1}\right)^2}{L_{x_1y_1}^2} + \frac{\left(R_{x_2y_2} - R_{x_1y_1}\right)^2}{R_{x_1y_1}^2}} \tag{14}$$

Therefore, the Euclidean distance of RSS pairs determines how much modeling error Okuli can tolerate in order to discriminate two locations. We refer to this tolerance level as *error bound*. A larger error bound means Okuli can tolerate larger modeling error, thus achieving higher spatial resolution. The error bound depends on both L_{xy} and R_{xy}, which renders it location-dependent.

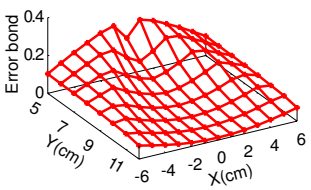

Figure 13: Modeling error bound at different locations.

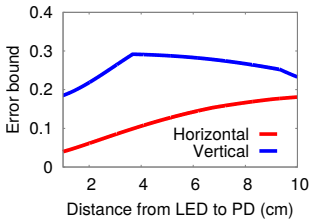

Figure 14: Error tolerance as system dimension varies.

In Figure 13, we plot the location-dependent distribution of error bound (maximum normalized error allowed to maintain 1cm accuracy), where we assume the angular responses of both LED and PDs have the typical cosine shape [12] and the workspace's dimension is consistent with the one used in our prototype. The separation between nearby spots to be localized is set to 1cm, along both horizontal and vertical directions. We observe a minimum error bound of 0.08, corresponding to the right hand side of Equation (14). From Equation (14), we can bound p and subsequently obtain the parameter k in Equation (13). For example, to achieve 90% localization confidence for the most error-prone location spot, the standard derivation of Okuli's modeling error σ should be bounded below 4%.

In practice, the error bound may differ from this analysis due to imperfect LED/PD response. However, the analysis can still be used empirically to determine how resilient Okuli can be for a given separation between target location spots.

System dimensioning. Based on the above analysis, we can numerically obtain *Okuli's localization resolution (i.e., error bound) as a function of the distance between LED and PD* (which affect the L_{xy} and R_{xy}). In Figure 14, we show the relationship between the minimum error bound and distance between LED and PD. The separation between spots is again set to 1cm. We can see the horizontal resolution keeps increasing with the LED-PD distance, *i.e.*, Okuli can tolerate more horizontal location errors with a larger LED-PD distance. This is because a larger distance between LED and PD could generate larger difference of angular response near the center which has the worst horizontal resolution. On the other hand, the vertical resolution has a turning point, since the sum distance from finger to left and right PD achieves maximum there. The vertical resolution is always higher than the horizontal one in practical range of dimensions (LED-PD distance from 1cm to 10cm).

As a result, in practical operating regime, increasing the distance between LED and PD is favorable. For instance, if the distance increases from 4cm to 10cm, the limiting error bound will nearly double, and resolution can be much higher.

5.4 Energy Efficient Light Emission

The power consumption of Okuli comes from three major components: LED, ADC and CPU, with the LED being the most power hungry one. Fortunately, Okuli can tame the LED energy cost using duty-cycling, *i.e.*, controlling the fraction of ON-period as it runs the OOK modulation (Section 5.1).

However, the duty-cycle cannot be arbitrarily reduced. Due to the rising/falling time of LED/PD and the settling time of ADC, a guard interval (GI) is needed between turning on/off the LED and reading RSS measurement from the ADC. The requirement of GI limits the updating frequency as well as the energy efficiency. In our prototype, the GI duration, denoted as t_{GI}, is large and equal to 15ms due to the slow rising/falling time of the PD. By definition of duty cycle, we have $D = t_{\mathrm{on}}/T$, where t_{on} is the duration of LED being turned on and T is the period (total duration of a cycle). Since we want to reduce the duty cycle, t_{on} should be set as short

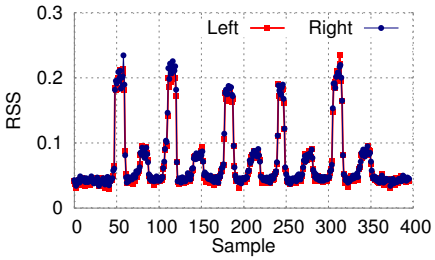

Figure 15: Time domain trace of RSS while alternately clicking a close-by and far-away point (relative to Okuli's front LED).

as possible, which implies $D \leqslant 0.5$ and $t_{\text{on}} = t_{\text{GI}}$ in the best case. Thus,

$$\frac{t_{\text{GI}}}{D} \leq \frac{t_{\text{on}}}{D} = T = \frac{1}{f}, \tag{15}$$

where D is the duty cycle and f is the sampling rate. Hence the minimum duty cycle D equals $t_{\text{GI}} \cdot f$, which is determined by the required GI and sampling rate, and cannot be further reduced.

We can model the system power consumption if we assume the power consumption of LED increases proportionally with duty cycle D and the power consumption of CPU increases proportionally with sampling rate:

$$\begin{aligned} P_{system} &= P_{static} + P_{\text{LED}} \cdot D + P_{\text{CPU}} \cdot f \\ &= P_{static} + P_{\text{LED}} \cdot f \cdot t_{\text{GI}} + P_{\text{CPU}} \cdot f \end{aligned} \tag{16}$$

where P_{static} is the static power consumption of the whole system. P_{LED} is the power consumption of LED when transmitting continuously and P_{CPU} is the power consumption of CPU while Okuli runs location update at 1 Hz. We can see a trade-off between sampling rate (*i.e.*, latency) and power consumption. Okuli can leverage this trade-off to adaptively switch to idle mode and wake up when finger is detected by changing the duty cycle. We leave the exploration of this mechanism for future work.

6. MOBILE INTERACTION USING Okuli

6.1 A Virtual Keyboard

We envision a virtual keyboard printed on a flat surface, or a cover that can be unfolded to one side of the Okuli-enabled mobile device. This virtual keyboard is more convenient and durable than a physical keyboard since it does not require any electrical connections or batteries. Finger clicks on the printed keys can be identified by Okuli, and interpreted by mobile applications. This requires us to augment Okuli with *finger-generated keystroke detection* along with *localization*.

Keystroke Detection. Unlike the aforementioned static finger localization, keystroke gestures generate highly dynamic light intensity at run-time — we need to isolate such dynamics from those caused by finger movement across locations. To meet this challenge, we take advantage of the unique time-domain signal features from keystrokes.

Figure 15 plots the RSS perceived by the left/right PDs during various keystroke events. We observe that the keystroke gestures generate a highly consistent pattern of RSS increasing and then sharply decreasing. This is partly due to the light grooming mechanism that constrains light reflection to a small slice of the finger, and partly due to the determinism of light propagation, which suffers relatively less multipath than RF signals [16].

This observation inspires us to use a simple threshold based feature detection algorithm to detect a keystroke event. During the initial 3-point calibration stage, when a user generates a *single keystroke* at the farthest point, the resulting 80-percentile RSS peak is used by Okuli as a barrier for declaring a keystroke. In addition, Okuli only passes peaks lasting more than 100ms in order to filter out spurious interferences.

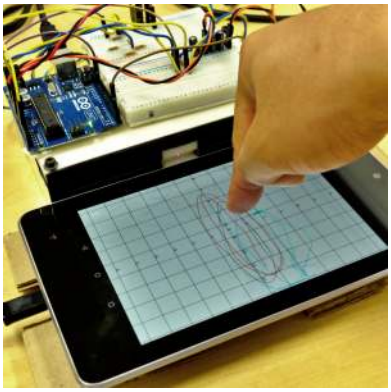

Figure 16: A portrait of Okuli implementation under test.

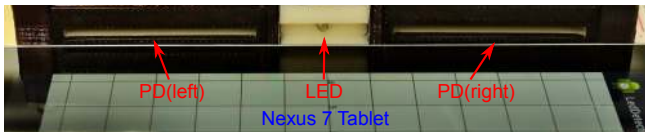

Figure 17: A close-up of Okuli implementation.

Keystroke Localization. Given a keystroke event, Okuli uses a two-step location matching algorithm to localize the key. It first computes the finger location by matching the resulting RSS to the model-predicted RSS of one 1mm×1mm grid on the workspace. It then maps the grid's location to the closest key (around 1cm^2 grid) based on the keyboard's geometrical outline. This approach is empirically proven to have superior performance than directly locating the keys (Section 8.2).

6.2 A Virtual Trackpad

Okuli can be easily extended to provide a trackpad-like experience for mobile interaction around a smartphone, smart watch, *etc.* A full-fledged trackpad entails two primitives: tracking finger movement and detecting clicks (taps).

Okuli extends its model-driven finger localization framework (Section 4) to realize tracking. It first obtains the millimeter-grid location estimation similarly to the keystroke localization. It then uses a simple averaging filter to remove outliers in the localization output. It keeps a moving average of a fixed number of recent localization results and averages their x and y coordinates to produce the final output. This technique removes abrupt jumps from trackpad traces, but also introduces a small delay in trackpad position updates. In our prototype of Okuli, a circular buffer of size 10 is used as the moving average window.

The click detection mechanism is naturally realized as part of the dynamic background removal (Section 5.2).

7. IMPLEMENTING Okuli

An experimental version of Okuli is implemented for research purposes (Figure 16 and Figure 17). It contains two PDs and an LED, a 3D-printed shroud to control the PDs' FOV, an embedded microcontroller to collect ADC samples from the PDs, a Nexus 7 Android tablet to display the output and run the localization algorithm, and a Serial-To-Bluetooth module to send samples from the embedded microcontroller to the Android tablet.

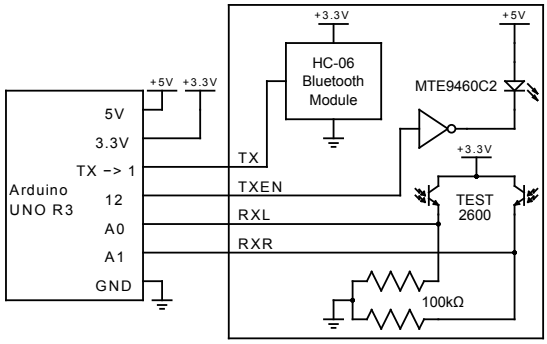

Figure 18: Simplified schematic of the Okuli experimental prototype.

Electronics. A number of factors are considered when choosing the experimental implementation's electronic components. The LED used for the experimental implementation is *MTE9460C2*, which has an FOV of around 120° in both horizontal and vertical directions. To conserve power, it would be better to have an LED custom designed with a narrower (30°) vertical FOV. This would allow a higher percentage of the LED's total light to be concentrated in useful directions and help alleviate multipath interference. The PD used is *TEST2600*, which has an FOV of nearly 120° in axial direction but 60° in the lateral direction. Due to such characteristic of the PD, we place the PDs with their axial direction being horizontal to maximize the FOV. The distance between the PDs is 8.8cm, and LED is slightly in front of the PDs. The shroud isolates the LED and the PDs so the light emitted by LED does not directly come into the PDs. Both the LED and the PDs work in the 950nm near-infrared spectrum, which is very close to visible light.

Okuli also needs a small amount of supporting electronic components. First, it needs a transistor to drive the LED and enable blinking, which is essential for ambient light removal and energy efficiency. Second, Okuli needs to digitize the light received from the PD's. To do this, we use an Arduino UNO with a built-in ADC. The Arduino sends the sampled light intensity values to a Nexus 7 via a Bluetooth interface. Figure 18 shows the schematic corresponding to this setup.

We emphasize that a production version of Okuli can be directly built into a mobile device, evading the Arduino/Bluetooth interface plus discrete circuit components. This will significantly reduce the form-factor and improve the energy-efficiency compared with our prototype.

Applications and UI. We implement the finger localization algorithm and the virtual keyboard/trackpad applications (Section 6) in Android. The Android program runs on a Nexus 7 tablet, which processes light intensity samples, detects touch events, maps the finger location to keystroke or writing trajectory, and displays the results on the screen.

Limitations. Due to limitations with our hardware prototype (discussed in Section 5.4), Okuli can only sample the PDs at 30Hz. Together with the 10-sample smooth filter, this produces a noticeable delay of 0.33 seconds for each localization output. In a production version of Okuli, much faster ADCs and PDs could be used, which would hide this delay and would allow lower-latency position updates.

8. SYSTEM EVALUATION

In this section, we first evaluate Okuli's performance in terms of modeling accuracy, finger-localization accuracy, and system energy efficiency. Then we showcase Okuli's effectiveness when serving virtual keyboard/trackpad applications.

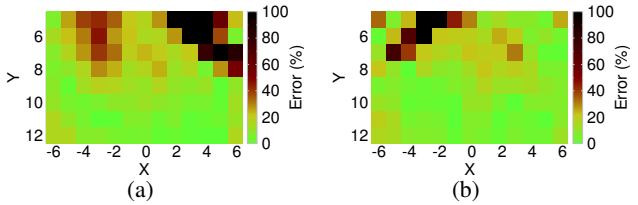

Figure 19: Modeling error within the workspace for (a) left PD and (b) right PD. X coordinate is parallel to the LED-PD line (Okuli's front face) and center-aligned with the LED. Y coordinate denotes the distance from the LED-PD line.

8.1 Microbenchmarks

8.1.1 Accuracy of Modeling Framework

We first evaluate the accuracy of the light propagation/reflection model (Section 4). Specifically, we follow Section 4.3 to perform the hardware and user calibration, and then directly compute the expected light RSS distribution in a $13\text{cm} \times 8\text{cm}$ workspace, with $1\text{cm} \times 1\text{cm}$ resolution. We compare this model output with actual RSS measurement when a finger is placed at each $1\text{cm} \times 1\text{cm}$ spot.

Figure 19 depicts the error map within the workspace. Each grid shows the percentage error of the model *w.r.t.* the spot-by-spot measurement. We observe that the modeling error is location dependent. First, close to the bottom of the workspace, there exists a $9\text{cm} \times 4\text{cm}$ region with high accuracy (90-percentile error below 10%), for both the left and right PDs. This region can reliably serve trackpad-like applications. In addition, some significant error exists, and are generally on a line pointed from PD/LED towards outside. This kind of radiative error pattern can be caused by errors introduced in angular response calibration. The factory calibration we have done is coarse-grained due to limited resolution of the light sensor we are using. The calibration tends to induce larger error near the outer-edge of the angular response curve where the LED/PD's gain tends to be small and not discernible. We expect such errors to be reduced in a stricter factory calibration for actual products.

Second, the modeling error tend to be slightly larger near the left/right edges, since the edges are further away from the LED/PDs, suffering from insufficient lighting. There also exists a few black spots, which is caused by the edge of our 3D-printed shroud blocking part of the LED signals. A product-level implementation with customized lenses would be able to overcome such imperfectness (Section 3).

Furthermore, note that areas with larger error tend to overlap with areas where the PDs have low angular response. Lower response means lower SNR, which causes larger error. The variation of angular response and propagation loss over space leads to location-dependent SNR and error tolerance (Section 5.3), which in turn explains the location-dependent error distribution.

8.1.2 Accuracy of Finger Localization

As analyzed in Section 5.3, Okuli can tolerate a certain amount of modeling error, as long as it does not confuse the RSS-vector at different locations. To test its localization accuracy, we focus on 100 randomly-picked points (including repeated locations) in a $9\text{cm} \times 7\text{cm}$ area (including points on the edge) within the workspace. All the points are selected on the $1\text{cm} \times 1\text{cm}$ grid. The workspace sits on a wood table surface in an office environment with natural background (fluorescent lighting and people walking by occasionally).

We test two different setup: the "manual" case localizes a static finger, while the "auto" case detects a finger touch and then localizes

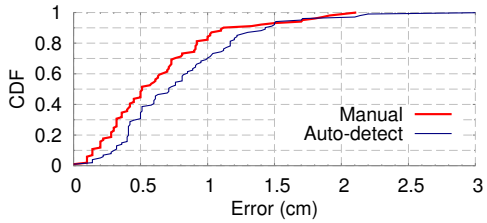

Figure 20: CDF of localization errors.

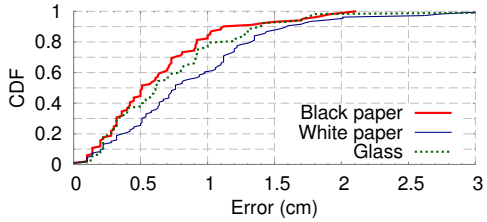

Figure 21: CDF of localization errors on various surfaces.

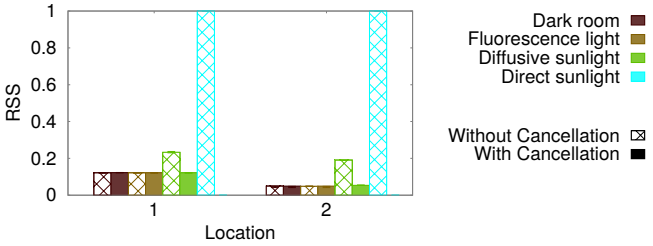

Figure 22: RSS under different lighting condition with and without ambient cancellation.

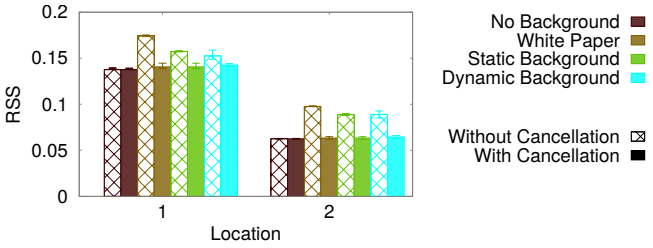

Figure 23: RSS under different background with and without dynamic background removal.

the finger. As shown in Figure 20, for the static finger localization, over 80% points have an error of less than 1cm, and 90% of the error values is below 1.12cm. When the location matching works in conjunction with finger detection, the error increases, because there exist decision errors that make it possible for the finger to become registered before it makes full contact with the surface, or is on its way leaving the surface. This will result in the reading being smaller than desired. Still, 90% of the errors fall below 1.43cm in this case. We have also conducted a closer examination of the spatial error distribution, and found the edge-spots tend to suffer from larger errors, which is consistent with the previous micro-benchmark.

It is also possible to use Okuli on various surfaces. As Figure 21 shows, the CDF of localization error remains relatively consistent on surfaces including black paper, white paper and a mirror-like glass. This corroborates the effectiveness of our model and calibration, which takes surface-specific impact on light reflection into account.

8.1.3 Effectiveness of Interference Suppression

Effectiveness of ambient light cancellation. To evaluate the performance of the ambient light cancellation, we put Okuli under various lighting conditions, and measure the RSS before/after the cancellation while putting finger on a fixed location. As shown in Figure 22, ambient light cancellation works ideally in an indoor environment with fluorescent light sources, whose signal strength is usually weaker or comparable with Okuli's near-field LED light.

To test Okuli under diffusive sunlight, we place it near the window facing towards inside of the building. Despite the strong ambient light interference (shown in case w/o cancellation), Okuli can restore its legitimate RSS after cancellation. The ambient light cancellation stops working under direct sunlight, especially on a sunny day, due to ADC saturation (Section 5.1). The problem may be alleviated by increasing the ADC's dynamic range or using analog cancellation, but this is left for our future work.

Effectiveness of dynamic background removal. To evaluate the performance of dynamic background removal, we arrange 4 different kinds of background: *(i)* very far-away background that can be treated as no background, *(ii)* a white paper standing vertically in front of Okuli, *(iii)* static background with the user himself/herself being the only significant background, and *(iv)* dynamic background with people walking by constantly. We then record the RSS before and after the cancellation. From Figure 23, we can see that although the RSS before cancellation varies significantly, it can be restored to the original level after cancellation. This indicates that our dynamic background cancellation is indeed effective.

Temporal stability of finger localization. It is a common concern for systems requiring initial calibration, that the usable duration of each calibration can be short, due to environmental changes.

To find out whether Okuli can perform consistently throughout a period of time with one calibration, we repeated the prior localization accuracy test twice – immediately after the system is calibrated and 10 days afterwards. As shown in Figure 24, the localization accuracy almost does not change over a long period of time. This is partly due to the determinism of the light propagation channel, and partly due to Okuli's capability to suppress environment dynamics throughput background cancellation.

8.1.4 System Energy Efficiency

To measure the power consumption of Okuli, we disconnect the Bluetooth module (since it is not needed in a product-level implementation), and use the Monsoon Power Monitor [17] to supply the Arduino and the sensor board with 4.5V power. We also use the same power monitor to measure the computational power consumption of our test App running on a Galaxy Nexus phone. In Figure 25, we show the power consumption of LED, CPU and ADC respectively. Consistent with our analysis in Section 5.4, the power consumption of LED dominates, and increases proportionally with the sampling duty-cycle. The CPU power follows a similar trend, but the absolute value is much lower. The ADC consumes a stable power of 35mW. Overall, at Okuli's default duty cycle of 0.5, the total system power consumption is 380mW, comparable to a typical web browsing session on a smartphone's WiFi interface [18].

Given the dominating LED power consumption in Okuli, we envision huge space for making Okuli more energy efficient. According to Equation (15), when the hardware can work with a short GI, it is possible to employ a duty cycle well below 0.5. For instance, if 100Hz sampling rate is desired, and the system can work with 0.5ms GI, then the minimum duty cycle is 0.05, hence the energy consumption can be reduce to around 100mW as predicted by Equation (16). Currently the GI of our system is limited by the hardware prototype which requires us to maintain a high duty cycle of 0.5 and hence relatively higher operating power than ideal.

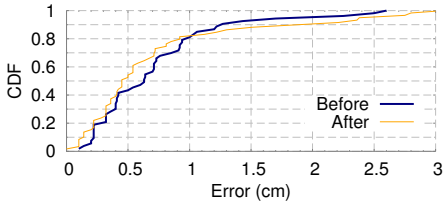

Figure 24: Localization accuracy before and after 10 days without re-calibration.

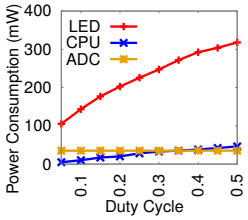

Figure 25: A breakdown of Okuli's power consumption.

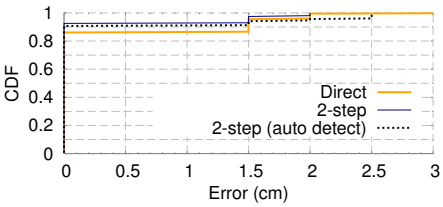

Figure 26: Localization error on a virtual keyboard.

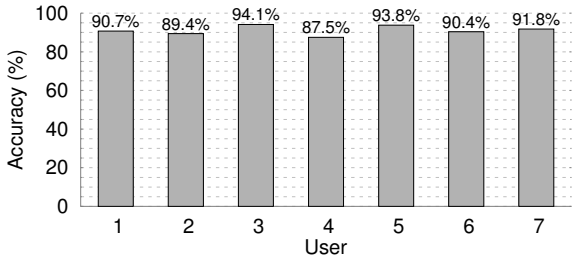

Figure 27: A user study of Okuli's virtual keyboard application.

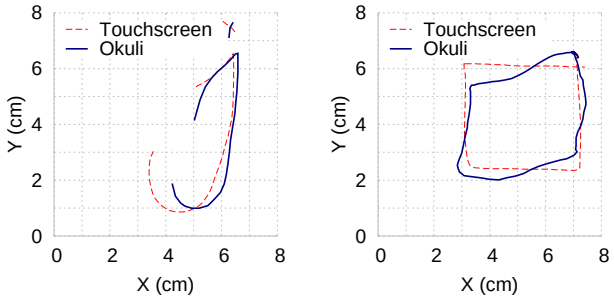

Figure 28: Okuli's virtual trackpad traces (solid) in comparison to touchscreen traces (dashed).

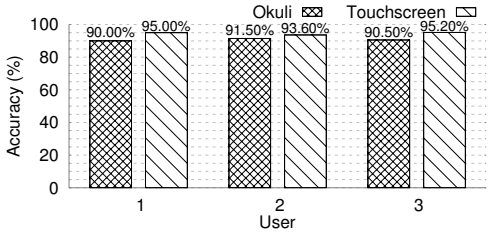

Figure 29: A user study of Okuli's virtual trackpad application.

8.2 Application Tests

Virtual Keyboard. We print 20 rectangular-shaped keys of 2cm width and 1.5cm height on a piece of paper, and use it as the workspace for Okuli's virtual keyboard application. The keys are next to each other and placed together in the center of the working area, arranged in a 4-key-per-row manner, and totaling 5 rows. After calibration, we generate 100 random keystrokes, and the error in distance of the pressed key and detected key is recorded. As shown in Figure 26, over 90% of the keystrokes are detected correctly with 2-step method, while the direct locate method only detects about 86% of the keystrokes correctly.

We also conduct a user survey that covers 7 users with different finger characteristics. Each user repeats the above experiment (20 1cm×1cm keys). The users are told to use the device just like a touchscreen. With 100 random key inputs on the same keyboard, average accuracy is 91.1%, and performance is highly consistent across users (Figure 27).

Virtual Trackpad. We evaluate the virtual trackpad's performance by replaying the user input captured into a handwriting recognition software, MyScript Stylus, to assess the accuracy for possible handwriting input applications. The user runs our virtual trackpad Android app to write on the Nexus 7 tablet, which is simultaneously used as the workspace for Okuli. Figure 28 shows two example writing trajectories.

We also conduct a user study of the trackpad, where each user inputs 100 random Latin characters. The result (Figure 29) demonstrates an overall recognition rate of 90.6%, which is comparable to writing directly on the Nexus 7 tablet (94.5%). The trackpad has a noticeable latency of around 0.5s due to the low-end PD, ADC and microcontroller we are using (Section 7). We expect a full-fledged implementation, integrated with a mobile device, will be able to overcome these limitations.

9. DISCUSSION

Multi-finger input and touch. Currently, Okuli is limited to single-finger localization, which has no ambiguity as long as the two PDs' angular responses are well conditioned (Section 5.3). By using more PDs, Okuli may further improve its resolution and discern multi-finger inputs. It would be interesting to investigate the system dimensioning and resolution as the number of PDs increases. However, this is beyond the scope of our current work.

Full-sized keyboard. Due to the PDs' limited field-of-view, the workspace of our current Okuli implementation cannot cover a full keyboard. However, this limitation can be easily overcome by using better hardware, *i.e.*, LEDs/PDs with wider FOV and higher sensitivity. In addition, since Okuli's localization mechanism is generic and can track fingers in arbitrary location within the working area, more complicated key arrangement other than the dial-pad-like layout is also possible.

System stability and usability. The light grooming mechanism and background suppression has greatly reduced the impact of irrelevant body parts and environment dynamics. Users' inconsistent behavior does affect the accuracy of the model behind Okuli. Fortunately, the range of such uncertainties is quite limited. In particular, we have observed in the user study that users' finger rotation/tilting is usually well below 60° unless intentionally induced. Moreover, Okuli's light grooming mechanism ensures a super-narrow vertical FOV, such that the PD still sees a relatively consistent light intensity even under such behavior dynamics.

Reducing processing latency. As a prototype, the circuit of Okuli is not optimally tuned at this stage. The photo-transistor is slower than a carefully chosen photo-diode, and there is no amplifier or matching circuitry between the sensors (which generate tiny photo-currents) and the microcontroller (which has a large input capacitance). We expect a full-fledged circuitry design can achieve significantly lower latency than what we observed in the prototype.

Impact of sunlight. The current Okuli implementation does not work well under direct sunlight. Sunlight creates very strong interference and easily saturates the sensors, rendering the ambient light cancellation ineffective. Yet, by employing hardware improvements, such as DC removal and high resolution ADC, this issue can be resolved.

10. RELATED WORK

Object location sensing technologies have been extensively explored for the purpose of human-computer/mobile interaction. The object can be human body, limb, hand, or as fine-grained as fingers. The underlying technologies can be classified into active sensing (*w.r.t.* objects instrumented with sensors or transceivers), and passive sensing (*w.r.t.* uninstrumented objects).

Visible light based *active location sensing* techniques have been driven by the emerging trend of visible light communication (VLC). A VLC link piggybacks information by flashing LED lights or an LED screen, which can be demodulated by a light sensor or camera [8, 19, 20]. The light attenuation vs. distance relation is explored in Epsilon [10] to complement motion sensors and locate a mobile VLC receiver. Luxapose [9] leverages multiple ceiling mounted LED lights to triangulate a smartphone, by using the camera and image distortion algorithms. Both systems achieve sub-meter scale location accuracy. Besides visible light, alternative medium of signals and devices have been explored in active sensing, including RFID [1], infrared [21], magnetic field [22] and motion sensors [2]. All these techniques require instrumenting the user with retro-reflective tags, sensors or transceivers.

Passive sensing systems can overcome such limitations. Infrared photoplethysmogram (PPG) tracks pulses or blood volume changes by measuring reflectance or transmittance. Infrared (IR) *proximity sensors* have also found mature applications in modern mobile devices. For example, SideSight [11] captures multi-touch gestures by observing the diverse proximity detection output of an array of 10 IR sensors, placed on the edge of a smartphone. Hoverflow [23] orients the IR sensor array towards the user to cover a large 3D space, allowing for handheld interaction. A similar concept was explored to facilitate gesture input for smart watch [24]. These systems can only sense coarse hand/finger gestures and require intensive training for each input. High-resolution, training-free multi-touch sensing has been demonstrated in ZeroTouch [25], FlexAura [26], *etc.*, which install hundreds of IR sensors around a flat panel or similar interaction space.

Okuli's fundamental design principle departs from these passive IR sensing systems, in that it treats the light propagation channel as a wide-angle diffusive channel, instead of a "laser-beam". Its fundamental approach is *model-driven* — it explicitly models the light intensity attenuation and reflection and can localize a finger within the wide FOV of an LED transmitter, instead of detecting finger "cutting" or proximity. This principle enables Okuli to achieve an unprecedented precision using only a single LED and two light sensors, allowing deployment on the edge of mobile devices of practical size.

Prior art has extensively explored vision based approaches for mobile interaction. Intuitively, by using a high-resolution camera instead of "single-pixel" photodetector, objects can be detected and localized with high precision. For example, hand gestures can be easily recognized via continuous video frame processing [27]. Infrared or near-infrared cameras can detect thermal emission or near-IR reflection from hands/fingers, thus identifying touch/click gestures [28–31]. Higher precision can be achieved via structured infrared-light projection [32], *i.e.*, emitting bar patterns or speckles towards target, and identifying its 3D structure and distance. This approach has been adopted by commercial hand/finger tracking systems like Kinect [33] and Leap Motion [34]. Such vision based systems, however, share many common limitations. As previously mentioned, they typically require a projector and/or multiple cameras, which do not suit very thin form factor devices. Also, cameras incur high energy cost under continuous sensing/processing, and hence they are unsuitable for battery-powered devices. Finally, cameras, especially side-facing ones, can pose threats to privacy if the system is compromised and exploited maliciously.

Passive sensing can be alternatively realized using acoustic sensors [5, 35], electric field sensors [36, 37], *etc.* These signals are relatively less deterministic compared with light, and thus they typically require extensive training before use.

11. CONCLUSION

Light emitters and sensors have been widely used to sense proximity or object presence, with the intuitive rationale of "object cutting light beams". In this paper, we have achieved a more ambitious goal of locating a small object on a 2D plane, with around one-centimeter scale precision, using only one LED emitter and two photodetectors (PDs). Our solution, called Okuli, uses a model-driven approach to localize a finger with minimal user training. We have navigated various design/performance tradeoffs in Okuli, using both analysis and real experiments. We believe Okuli marks an important step in passive object/activity sensing through visible light.

Acknowledgement

We appreciate insightful comments and feedback from the shepherd and the anonymous reviewers. The work reported in this paper is supported in part by the NSF under Grant CNS-1318292, CNS-1343363 and CNS-1350039.

12. REFERENCES

[1] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual Touch Screen in the Air using RF Signals," in *Proc. of ACM SIGCOMM*, 2014.

[2] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using Mobile Phones to Write in Air," in *Proc. of ACM MobiSys*, 2011.

[3] H. Roeber, J. Bacus, and C. Tomasi, "Typing in Thin Air: The Canesta Projection Keyboard - a New Method of Interaction with Electronic Devices," in *ACM CHI Extended Abstracts*, 2003.

[4] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: Wearable Multitouch Interaction Everywhere," in *Proc. of ACM UIST*, 2011.

[5] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous Keyboard for Small Mobile Devices: Harnessing Multipath Fading for Fine-Grained Keystroke Localization," in *Proc. of ACM MobiSys*, 2014.

[6] C. Harrison, J. Schwarz, and S. E. Hudson, "TapSense: Enhancing Finger Interaction on Touch Surfaces," in *Proc. of ACM UIST*, 2011.

[7] H. Elgala, R. Mesleh, and H. Haas, "Indoor Optical Wireless Communication: Potential and State-of-the-Art," *IEEE Communications Magazine*, vol. 49, no. 9, 2011.

[8] S. Rajagopal, R. Roberts, and S.-K. Lim, "IEEE 802.15.7 Visible Light Communication: Modulation Schemes and Dimming Support," *IEEE Communications Magazine*, vol. 50, no. 3, 2012.

[9] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proc. of ACM MobiCom*, 2014.

[10] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A Visible Light Based Positioning System," in *Proc. of USENIX NSDI*, 2014.

[11] A. Butler, S. Izadi, and S. Hodges, "SideSight: Multi-Touch Interaction Around Small Devices," in *Proc. of ACM UIST*, 2008.

[12] Z. Ghassemlooy, W. Popoola, and S. Rajbhandari, *Optical Wireless Communications: System and Channel Modelling with MATLAB*. CRC Press, 2012.

[13] A. Edelstein and M. Rabbat, "Background Subtraction for Online Calibration of Baseline RSS in RF Sensing Networks," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 12, pp. 2386–2398, Dec 2013.

[14] W. Cui, Z. Cao, and J. Wei, "Dual-Microphone Source Location Method in 2-D Space," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[15] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[16] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu, "Leveraging Directional Antenna Capabilities for Fine-grained Gesture Recognition," in *ACM UbiComp*, 2014.

[17] Monsoon Solutions, Inc., "Monsoon Power Monitor," http://www.msoon.com/LabEquipment/PowerMonitor/.

[18] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *Proc. of the USENIX Annual Technical Conference*, 2010.

[19] T. Hao, R. Zhou, and G. Xing, "COBRA: Color Barcode Streaming for Smartphone Systems," in *Proc. of ACM MobiSys*, 2012.

[20] S. D. Perli, N. Ahmed, and D. Katabi, "PixNet: Interference-free Wireless Links Using LCD-camera Pairs," in *Proc. of ACM MobiCom*, 2010.

[21] M. Ogata, Y. Sugiura, H. Osawa, and M. Imai, "iRing: Intelligent Ring Using Infrared Reflection," in *Proc. of ACM UIST*, 2012.

[22] C. Harrison and S. E. Hudson, "Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices," in *Proc. of ACM UIST*, 2009.

[23] S. Kratz and M. Rohs, "Hoverflow: Exploring Around-device Interaction with IR Distance Sensors," in *Proc. of ACM MobileHCI*, 2009.

[24] J. Kim, J. He, K. Lyons, and T. Starner, "The Gesture Watch: A Wireless Contact-free Gesture Based Wrist Interface," in *Proc. of IEEE International Symposium on Wearable Computers (ISWC)*, 2007.

[25] J. Moeller and A. Kerne, "ZeroTouch: An Optical Multi-touch and Free-air Interaction Architecture," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2012.

[26] S. Liu and F. Guimbretière, "FlexAura: A Flexible Near-Surface Range Sensor," in *Proc. of ACM UIST*, 2012.

[27] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based Hand-gesture Applications," *Communications of the ACM*, vol. 54, no. 2, 2011.

[28] A. Sahami Shirazi, Y. Abdelrahman, N. Henze, S. Schneegass, M. Khalilbeigi, and A. Schmidt, "Exploiting Thermal Reflection for Interactive Systems," in *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2014.

[29] E. Larson, G. Cohn, S. Gupta, X. Ren, B. Harrison, D. Fox, and S. Patel, "HeatWave: Thermal Imaging for Surface User Interaction," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2011.

[30] P. Mistry and P. Maes, "Mouseless: a computer mouse as small as invisible," in *Adjunct Proceedings of ACM UIST*, 2010.

[31] D. Kim, S. Izadi, J. Dostal, C. Rhemann, C. Keskin, C. Zach, J. Shotton, T. Large, S. Bathiche, M. Niessner, D. A. Butler, S. Fanello, and V. Pradeep, "RetroDepth: 3D Silhouette Sensing for High-precision Input on and Above Physical Surfaces," in *Proc. of ACM Conference on Human Factors in Computing Systems (CHI)*, 2014.

[32] R. Xiao, C. Harrison, K. D. Willis, I. Poupyrev, and S. E. Hudson, "Lumitrack: Low Cost, High Precision, High Speed Tracking with Projected M-sequences," in *Proc. of ACM UIST*, 2013.

[33] Microsoft Corporation, "Kinect for Windows," 2014. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/

[34] Leap Motion, Inc., "Leap Motion: Mac & PC Gesture Controller for Game, Design and More," 2013. [Online]. Available: https://www.leapmotion.com/

[35] C. Harrison, D. Tan, and D. Morris, "Skinput: Appropriating the Body As an Input Surface," in *Proc. of SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2010.

[36] M. Le Goc, S. Taylor, S. Izadi, and C. Keskin, "A Low-cost Transparent Electric Field Sensor for 3D Interaction on Mobile Devices," in *Proc. of ACM Conference on Human Factors in Computing Systems (CHI)*, 2014.

[37] T. Grosse-Puppendahl, S. Herber, R. Wimmer, F. Englert, S. Beck, J. von Wilmsdorff, R. Wichert, and A. Kuijper, "Capacitive Near-field Communication for Ubiquitous Interaction and Perception," in *Proc. of ACM UbiComp*, 2014.