

# LumiWatch: On-Arm Projected Graphics and Touch Input

Robert Xiao<sup>1</sup> Teng Cao<sup>2</sup> Ning Guo<sup>2</sup> Jun Zhuo<sup>2</sup> Yang Zhang<sup>1</sup> Chris Harrison<sup>1</sup>

<sup>1</sup> Carnegie Mellon University  
Human-Computer Interaction Institute  
5000 Forbes Ave, Pittsburgh, PA 15213  
{brx, yang.zhang, chris.harrison}@cs.cmu.edu

<sup>2</sup> ASU Tech Co. Ltd.  
8<sup>th</sup> Floor, Lunyang Tower  
6 North 3<sup>rd</sup> Ring Road Middle, Beijing, China 100011  
{bjch001, guoning0067, zhuojun0097}@a-su.com.cn

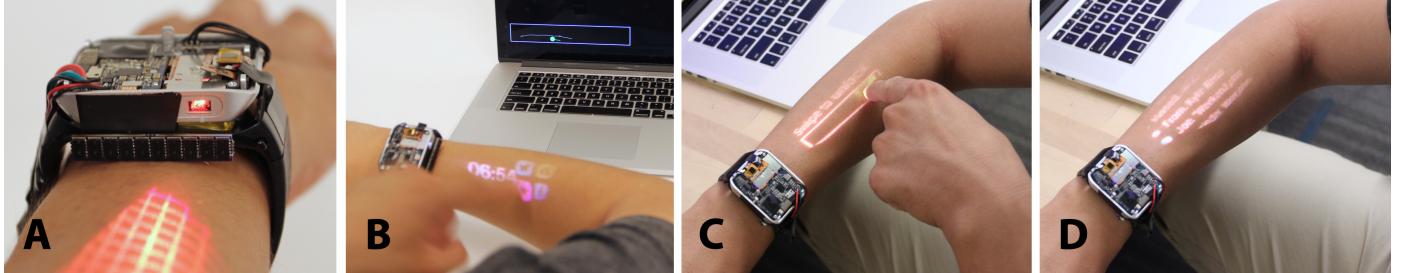


Figure 1. Our self-contained projection smartwatch (A) provides rectified graphics with touch input on the skin (B). We use a slide-to-unlock mechanism to reject inadvertent touches and provide a rapid projection calibration (C) before apps can be used (D).

## ABSTRACT

Compact, worn computers with projected, on-skin touch interfaces have been a long-standing yet elusive goal, largely written off as science fiction. Such devices offer the potential to mitigate the significant human input/output bottleneck inherent in worn devices with small screens. In this work, we present the first, fully-functional and self-contained projection smartwatch implementation, containing the requisite compute, power, projection and touch-sensing capabilities. Our watch offers roughly 40 cm<sup>2</sup> of interactive surface area – more than five times that of a typical smartwatch display. We demonstrate continuous 2D finger tracking with interactive, rectified graphics, transforming the arm into a touchscreen. We discuss our hardware and software implementation, as well as evaluation results regarding touch accuracy and projection visibility.

## Author Keywords

Smartwatch; projection; touch interaction; depth sensing; time-of-flight; on-body interaction.

## ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g. HCI): User interfaces: Input devices and strategies.

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 ACM 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.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5620-6/18/04...\$15.00

<https://doi.org/10.1145/3173574.3173669>

## INTRODUCTION

Appropriating the human body as an interactive surface is attractive for many reasons. Foremost, skin provides a natural and immediate surface for dynamic, digital projection. Although it introduces some color and physical distortion, the resolution, framerate and overall quality can be high [14, 16, 30, 49]. More importantly, it offers considerable surface area for interactive tasks – many times that of e.g., a smartwatch display. With today’s smartwatches containing multi-core, multi-gigahertz CPUs, one could argue their small touchscreens are the chief bottleneck to unlocking richer and more useful applications. Indeed, several widely publicized, *conceptual* on-skin projection watches have been proposed, most notably Cicret [6] and Ritot [37].

A second benefit is that our bodies are always with us, and are often immediately available [39, 45]. This stands in contrast to conventional mobile devices, which typically reside in pockets or bags, and must be retrieved to access even basic functionality [2, 17, 38]. This generally demands a high level of attention, both cognitively and visually, and can be socially disruptive. Further, physically retrieving a device incurs a non-trivial time cost, and can constitute a significant fraction of a simple operation’s total time [1].

Lastly, as the colloquialism “like the back of your hand” suggests, we are intimately familiar with our own bodies. Though proprioception, we can easily navigate a finger to our palm, even with our eyes closed. We have finely tuned muscle memory and hand-eye coordination, providing a high level of input performance, for both absolute touch location and relative gesturing – powerful interaction modalities that worn systems can leverage.

However, despite these significant benefits, building practical, worn projection systems has remained elusive. In order to achieve sufficient projection brightness, sensing robust-

ness and compute power, past on-body systems have employed full-sized components (*e.g.*, USB depth cameras [14, 43], portable projectors [30, 49]). The resulting size of these systems mean that they must be worn on the upper arm or shoulder, and most often tethered for compute and power [14, 16, 29, 30, 49].

We present *LumiWatch*, a custom, tightly integrated and fully self-contained on-skin projection wristwatch. It incorporates a 15-lumen scanned-laser projector, a ten-element time-of-flight depth-sensing array, quad-core CPU running Android 5.1, and battery good for one hour of continuous projector operation (or one day of occasional use). Our prototype measures 50×41×17 mm, nominally larger than the production 42 mm Apple Watch Series 3 (43×36×11 mm). With our watch, we demonstrate continuous 2D finger touch tracking on the skin with coordinated interactive graphics. Owing to the shallow angle of projection, our graphics pipeline must rectify interfaces to the complex, non-planar geometry of the arm.

Our hardware and software, taken together, transform the arm into a coarse touchscreen, offering roughly 40 cm<sup>2</sup> of interactive surface area, more than five times that of a typical smartwatch. This interactive area supports common touchscreen operations, such as tapping and swiping, allowing it to offer similar interactions to that of a (single-touch) smartphone. Although obstacles remain for practical adoption, we believe our work demonstrates the first functional projection smartwatch system and constitutes a significant advance in the state of the art.

## RELATED WORK

The primary goal of on-body interfaces is to provide “always-available input”, where a user does not need to carry or pick up a device [39, 45]. To support this class of interactions, numerous approaches have been considered. The most straightforward is to take conventional physical computing elements and place them on the body; iconic examples include a one-handed keyboard [26] and a wrist-bound touchpad [46]. Integrating input capabilities into clothing has also been the subject of considerable research [28, 36].

More related to our present work are techniques that attempt skin-based touch tracking from an arm-worn wearable. We also discuss a much smaller body of work looking at on-body projection, including a few systems that achieved both input and output on the body.

### On-Skin Touch Input

Given that the surface area of one hand exceeds that of a typical smartphone screen, and humans have roughly 1-2 m<sup>2</sup> of skin surface area in total, it is unsurprising that researchers have tried to appropriate it for digital input. A wide variety of sensing approaches have been considered, which we briefly review.

Optical approaches are most common. For example, SenSkin [33] and SkinWatch [34] both used infrared proximity sensors to detect skin deformations resulting from touch

inputs. More similar to our system, Lim et al. [24] used a pair of infrared emitters and a photodiode array attached to the side of a watch to track a 2D finger location, while SideSight [5] used an array of proximity sensors to sense finger position around a phone. Researchers have also investigated worn cameras for touch tracking, including arm [43] and shoulder [7, 13] vantage points.

Acoustic methods are also powerful. SonarWatch [23] and PUB [25] both used ultrasonic sonar to measure distance to a finger interacting on an arm’s surface, providing 1D touch tracking. It is also possible to instrument the finger with an active signal source, as seen in The Sound of Touch [31]. There are also passive techniques, such as TapSkin [50] and ViBand [22], which rely on bioacoustic propagation of vibrations resulting from taps to the skin.

There has also been some work in RF and capacitive sensing, for instance, Touché [41] used a wrist-worn, swept frequency capacitive sensor to detect gestural interactions between a wearer’ two hands. The instrumented smartwatch seen in AuraSense [52] used projected electric fields to enable close-range buttons and sliders on the skin. By using an active ring, Skintrack [51] enabled continuous 2D finger tracking on the skin through RF triangulation. Finally, the skin itself can be instrumented, sidestepping many remote sensing issues. DuoSkin [18], iSkin [47] and Kramer et al. [20] have all demonstrated flexible, skin-compatible overlays supporting capacitive touch sensing.

### On-Skin Projected Interfaces

Unsurprisingly, the art community was among the first to embrace the fusion of projected media and the human form. Examples include the opening sequence to Guy Hamilton’s “Goldfinger” (1964) and Peter Greenway’s “The Pillow Book” (1996), both of which projected imagery onto actors’ bodies for dramatic effect. More recently, an interactive installation by Sugrue [44] allowed visitors to touch a screen containing virtual “bugs” that would move out onto people’s hand and arms via overhead projection. Barnett [3] provides a survey of many of these artistic efforts.

Owing to the size, weight and power consumption of computers and projectors, it is most common to find on-body projection systems installed in the environment. For example, TenoriPop [32] rendered interactive graphics onto the hands of shoppers using a ceiling mounted projector/camera rig. Similar overhead setups have also been considered for medical uses, where *e.g.*, anatomy can be overlaid onto bodies for surgical assistance [11] and education [8, 35]. In the HCI literature, LightSpace [48], LightGuide [42] and Armura [15] all used overhead setups to enable a variety of on-body projected interactions.

Rarest are *worn* systems that attempt both input and graphical output on the body. This is a nascent, but growing literature, often referred to as “on-body interfaces”. Early systems include Sakata et al. [40], which describes a “palm top” projection system using fiducial markers worn on the

wrist to provide 6DOF position tracking of a wearer’s hand. Similarly, SixthSense [30] tracked color markers worn on the fingers with a neck-worn camera to detect finger input; an integrated pico-projector could render interfaces onto the body or environment.

Other work has aimed to avoid instrumenting users with markers. For example, Skinput [16] – worn on the upper arm – relied on bioacoustic signals resulting from touches to the skin. Continuous finger tracking was not possible, so projected interfaces were built around pre-learned touch locations. Using computer vision, PALMbit [49] could track finger-to-finger touches without markers, enabling a projected interface on the palm. Finally, OmniTouch [14] used a shoulder-worn computer vision system to track free-form multitouch finger inputs on the body and environment.

Of note, none of the above systems attempted touch tracking and projection in a smartwatch-like device, as the small size requirement, shallow-angle projection, and oblique sensing viewpoint all pose significant challenges. The closest system to this desired form factor is [21], which used fixed-icon (*i.e.*, rendered to film) laser projections coupled with IR proximity sensors to detect clicks to “skin buttons”.

### LUMIWATCH HARDWARE

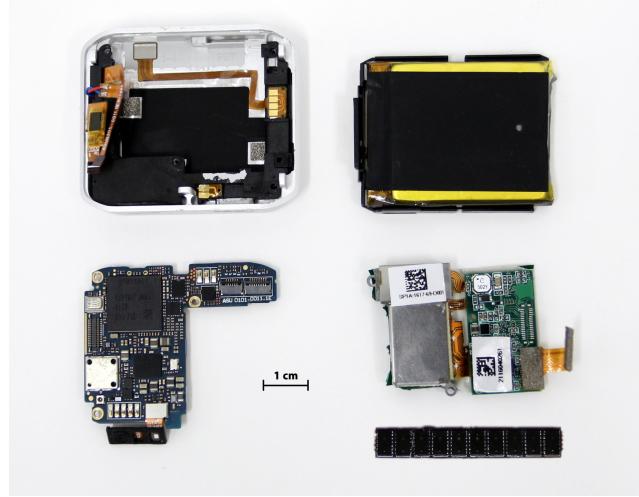
Our custom smartwatch hardware consists of five primary components, seen in Figure 2: a logic board, projector, depth-sensing array, metal enclosure and battery. It is fully self-contained and capable of independent operation (*i.e.*, no tether to a smartphone or computer). We estimate immediate retail cost would be around \$600.

#### Logic Board

Our smartwatch logic board (Figure 2, bottom left) was designed around a Qualcomm APQ8026 system-on-chip, which integrates a 1.2 GHz quad-core CPU, 450 MHz GPU, and Bluetooth 4.0 and WiFi controller. We also added 768 MB of RAM, 4 GB of flash memory, inertial measurement unit (IMU) and ambient light sensor. The component placement on the logic board was optimized to reduce thermal interference from our projector. Our smartwatch runs Android 5.1, with smartwatch-specific applications.

#### Projector

We designed and manufactured a custom 15-lumen pico-projector module for our smartwatch. The projector uses three lasers (red, green and blue) with a pair of MEMS mirrors operating in a raster-scan mode (*i.e.*, a scanned laser design [19]). This emits a  $1024 \times 600$  image at 60 Hz across a  $39^\circ \times 22.5^\circ$  field of view (Figure 3). Our projector module measures  $25.8 \times 16.6 \times 5.2$  mm (Figure 2, bottom-right), and consumes up to 2.7 W of power displaying a maximum-brightness, full-white image. With scanned-laser designs, power usage varies by content, *e.g.*, black imagery requires almost no power. The projector module is paired with custom drive electronics that controls the lasers and mirrors, while exposing a standardized display interface to the Android firmware.



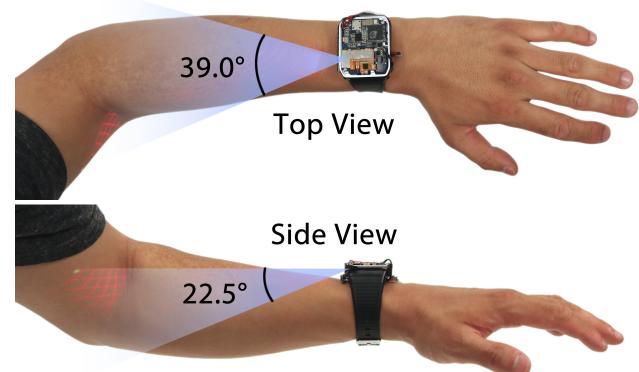
**Figure 2. Main hardware components of LumiWatch.** Clockwise from top-left: heat-dissipating aluminum shell; battery; projector module and driver; 1D sensor array; and logic board.

#### Depth-Sensing Array

To capture touch input on the projected area, we designed a compact 1D depth-sensing array ( $7 \times 38 \times 3$  mm), consisting of ten STMicro VL6180X time-of-flight sensors (Figure 2, bottom-right). Each individual sensor determines the distance to the nearest object within a  $25^\circ$  cone by emitting pulsed infrared light and timing the returned reflection. These sensors are connected to a dedicated microcontroller (NXP MK20DX256) over I<sup>2</sup>C, which triggers measurements in round-robin fashion to avoid infrared collisions. Capturing all ten measurements takes  $\sim 36$ ms, resulting in a touch tracking frame rate of 27.5 Hz.

#### Battery & Shell

The smartwatch electronics are contained in an aluminum shell (Figure 2, top-left), which assists in dissipating heat generated by the projector and logic board. The entire smartwatch prototype (logic board, projector, and depth sensor) is powered from a 740 mAh, 3.8 V (2.8 Wh) lithium-



**Figure 3. Top and side view of LumiWatch, with illustration of projector’s field of view. Top: tangential field of view. Bottom: axial field of view.**

um-ion battery (Figure 2, top-right), making it fully self-contained. The battery sits below the projector, close to the skin. Combined with the thickness of the enclosure, this causes the center of the projector aperture to sit approximately 13 mm above the surface of the skin, providing a slightly higher vantage point for projection. Under average use conditions, we obtain over one hour of continuous projection (with CPU, GPU, WiFi, and Bluetooth all active). In more typical smartwatch usage, where the projection would only be active intermittently, we expect our battery to last roughly one day.

### TOUCH TRACKING

We use our 1D depth-sensing array to track fingers on or very near to the surface of the arm. The conical fields of view of each individual time-of-flight sensor overlap significantly, so that a single finger will always appear in several sensors simultaneously (Figure 4). However, the finger will always be closest to the sensor most directly inline with the sensing axis, allowing us to infer a finger's tangential  $y$  position on the arm (in addition to the axial  $x$  position along the arm provided by the distance measurement itself). Thus, our 1D depth-sensing array gives us the ability to track the 2D position of a finger on the arm.

Our finger-tracking algorithm works in stages. First, the incoming sensor values are smoothed with an exponentially-weighted moving average (EWMA) filter to reduce noise and jitter (Figure 4 top, red dots). Next, sensors with invalid measurements or distances  $> 150$  mm are removed from consideration. If there are at least two sensors remaining, the algorithm detects a finger. It uses the minimum distance measurement among all sensors as the raw  $x$  position of the touch point. Then, to find the  $y$  position, it computes a weighted average of the reported sensor positions:



**Figure 4. Top of screen: raw data from our ten time-of-flight sensors (red dots), with estimated touch point shown in green. Bottom of screen: resulting touch paths. On arm: current path is projected for debugging.**

$$w_i = \frac{1}{x_i - x + d}, \quad y = s \cdot \frac{\sum_{i=1}^{10} i w_i}{\sum_{i=1}^{10} w_i}$$

where  $d$  is a tuning parameter that mitigates noise (set to 5 mm in our implementation) and  $s$  is the pitch between sensors (here,  $s = 3.78$  mm). This average heavily weights  $x_i$  values near the minimum  $x$ , thus effectively allowing it to average between multiple noisy minima while ignoring values far from the minimum. Finally, the  $x$  and  $y$  values are smoothed with an EWMA filter and reported to running applications.

On top of the basic  $x$  and  $y$  finger position, we detect single finger taps and swipes, to enable richer application input. Both detection mechanisms operate on *finger strokes*, which consist of the entire trace of a finger's position from initial detection to disappearance. A short finger stroke (total distance travelled less than 20 mm) is inferred as a finger tap, while long finger strokes are considered swipes. The position of a finger tap is taken as the point of minimum velocity during a short stroke. The swipe orientation (horizontal or vertical) is determined by comparing the total  $x$  and  $y$  travel distances across the long stroke; a stroke that travels at least 3 times further in the  $x$  direction as the  $y$  direction is considered horizontal. This factor of 3 accounts for the proportions of the arm, as strokes along the arm were observed to be significantly longer than vertical ones.

### PROJECTED OUTPUT

Most digital projectors emit a rectangular image that ideally falls on to a flat, perpendicular surface. However, from the perspective of our smartwatch, an arm's surface is neither flat or perpendicular, but rather irregularly curved and nearly parallel. This introduces a number of challenges, which we address through a combination of hardware design and efficient software calibration. Our graphical rectification pipeline is lightweight enough to allow application software to run at our projector's native 60 frames per second.

### Hardware Design

First, the hardware is designed to elevate the projector as high as possible without wasting space. To do this, we made the projector the top most component in our watch, allowing the aperture to lie 13 mm above the surface of the skin. At this height, a pixel 150 mm from the projector on the arm's surface is less than 5° from parallel. A shallower projection angle results in larger pixels and coarser projection, as well as increased sensitivity to variations in arm angle and geometry.

To keep the smartwatch thin, the projector is mounted so that the shorter 600-pixel axial projection axis extends horizontally across the skin, with the field of view extending from 0° horizontal at pixel 600 (parallel to an imaginary flat arm) down to 22.5° below the horizontal at pixel 0 (pointing down towards the arm). See illustration in Figure 3. This results in a gap of approximately  $13 \cot(22.5^\circ) = 31$  mm between the projector and the projected image on the arm. Here, we note that increasing the field of view of the pro-

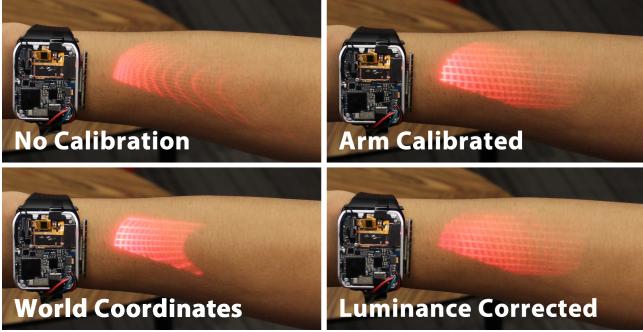


Figure 5. Various stages of projection calibration.

jector will slightly reduce this gap, but will have a significant negative impact on the fidelity of the image far from the projector (*i.e.*, larger pixels). Thus, increasing the  $y$ -axis field of view is not necessarily desirable for a projection smartwatch.

#### World Coordinate Rectification

Figure 5, top-left, shows the default projection without any correction (*i.e.*, projecting as if onto a perpendicular flat plane). The first software correction step is to achieve projection in “world coordinates,” that is, determining the mapping between the projector’s image coordinates and real-world 3D coordinates. These world coordinates also correspond to touch input positions, making this correction doubly crucial. As this calibration depends only on the hardware alignment between the projector and touch sensor, it can in practice be performed “at the factory.” In our system, we calibrate by finding the projector pixel coordinates corresponding to ten non-coplanar points in the real world, and then use a least-squares optimization algorithm to compute the projection matrix which best maps these 3D points to the 2D projector coordinates. With this projection matrix, we can specify draw coordinates in world coordinates, and the resulting projected pixels will appear at those coordinates. Notably, our smartwatch’s GPU performs this correction with no additional computational load because rendering with a projection matrix is already performed by default in a 3D context, making this a very efficient calibration.

#### Arm Calibration

On its own, this “world correction” would be sufficient to display on a flat shallow-angle surface, such as a table. However, the arm is not a flat surface, but rather an irregular cylinder-like object. The curvature of the arm introduces severe distortions relative to a flat surface (Figure 5, bottom-left), and so the next step is to calibrate to the arm. This stage depends on the wearer’s arm geometry. When this information is not available, a “default” calibration for an average arm can be used, which will still be significantly superior to no arm calibration at all (Figure 5, bottom-left vs. top-right).

Ideally, arm calibration is performed using a 3D model of a user’s arm. However, we created a simple model by interpolating between two ellipses representing the user’s wrist and forearm cross-sections. We render all application con-

tent to an offscreen texture, then map the texture onto the model, effectively performing projection mapping onto the arm. This rendering process removes the curvature of the arm (Figure 5, top-right). Rendering a textured 3D model is quite efficient using our onboard GPU, so this calibration step can be performed with minimal overhead.

#### Luminance Correction

Next, we perform luminance correction, as the pixels near to the projector are brighter than those farther away (as seen in Figure 5, top-right). This is achieved by pre-computing a map of the approximate distance from the projector to each pixel on the arm. This map is then transformed into a luminance correction map by using the relation:

$$\text{factor} = \text{clamp}(\text{dist}^2 / \text{maxdist}^2, 1/16, 1)$$

This configures a correction which scales the luminance by the square of the distance, but which is clamped to avoid reducing the luminance too much for close pixels; *maxdist*, which we set to 70 mm, configures the furthest distance to apply the correction, beyond which pixels are displayed at full brightness. Put simply, this correction reduces the brightness of the pixels closest to the projector, providing a more uniform appearance (Figure 5, bottom-right).

#### DYNAMIC PROJECTION-INPUT CALIBRATION

The prior projection calibration steps were all fixed steps that could be performed once per user or device, and then saved for future use. However, the precise angle between a wearer’s wrist and forearm cannot be calibrated *a priori*, because this angle can differ each time the watch is worn or wearer lifts their arm, necessitating some kind of dynamic calibration. For example, a difference of just one degree in the tilt of the projector can produce significant shifts in the projected output – *e.g.*, a pixel that appears 130 mm away with the arm at 0° would appear at 110 mm with the arm angled just 1° towards the projector.

To solve this problem, we leverage the familiar *swipe to unlock* gesture often used on smartphones. Upon lifting the smartwatch to initiate an interaction, the user is prompted with a “swipe to unlock” slider (Figure 1C). The slider begins at a constant axial distance of 70 mm based on the default arm angle, but may appear in a different place depending on a user’s true arm angle. When a user “grabs” the slider’s handle, the finger’s axial position is recorded, and the true arm angle is recovered by comparing the actual axial position with the assumed position of the slider and applying trigonometry. In a similar fashion, the touch points generated by the swipe allow us to dynamically calibrate our world coordinate transform. Upon completion of the swipe gesture, the recovered calibrations are applied for the remainder of that interactive session (*i.e.*, until the user stops interacting with the watch).

Thus, this simple unlock gesture serves three purposes: 1) it allows the arm angle to be seamlessly and intuitively calibrated without an explicit calibration routine, 2) it provides the necessary dynamic calibration for aligning the projector

and touch sensor, and 3) it verifies the user's intent to interact with the system (mitigating *e.g.* false inputs).

## EVALUATION

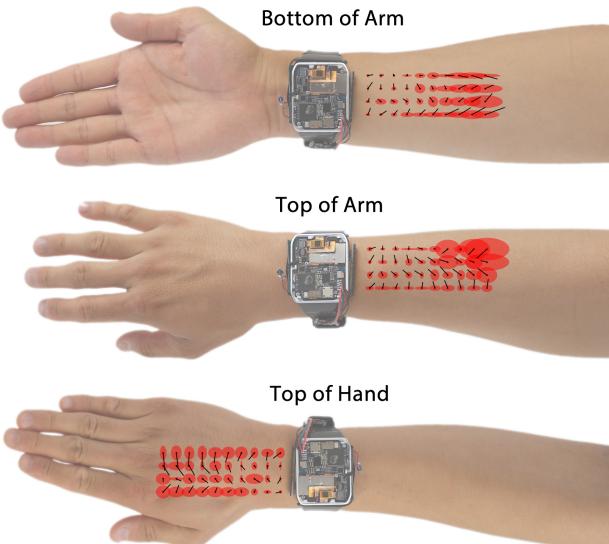
We performed two focused studies to evaluate the performance of our system. The first study evaluates input performance, while the second tests projection performance.

### Study 1: Input Performance

We recruited five participants (one female, two left-handed) to evaluate the touch input performance of our system, who were compensated \$20 USD for the 30-minute study. We measured the wrist and forearm circumferences for each user, which ranged between 133~168 mm and 197~302 mm respectively. Participants wore the watch on the arm opposite their dominant hand in three different projection positions (tested in random order): bottom of arm, top of arm, and top of hand (illustrated in Figure 6). Note that we did not test input performance on the palm because the hypothenar and thenar eminences occluded sensing and projection. Participants sat in a chair and held their arm up as if to check the time on a wristwatch.

For each projection position, users performed a *tap* task and a *swipe* task. Before the tap task, the experimenter used a flexible stencil to draw a 4×10 grid of dots, spaced 10 mm apart, extending outwards from the watch. The user was instructed to touch their finger to each dot in a sequential order. The system ran the tap detector in real-time, recording any tap event received. After each touch, the experimenter would click a key to record the trial (even if no touch was registered by the system).

In the swipe task, users performed four different swipe gestures (left, right, up, and down) ten times each in a random



**Figure 6. Our three tested projection locations with an illustration of results from the touch study. Red ellipses are sized to  $\pm 1$  standard deviation, black lines represent mean displacement across all users.**

order. Participants were instructed to perform a swipe direction, after which the experimenter clicked a key to advance to the next trial. The system ran the swipe detector in real-time, recording each gesture detected.

In total, we collected 600 touch points (40 dots × 3 projection locations × 5 participants) and 600 swipe gestures (4 swipe directions × 10 repeats × 3 projection locations × 5 participants).

### Touch Accuracy Results

There were no significant differences in tap accuracy across the three projection locations. Over all 600 trials, a tap was detected in 596 of them, for an aggregate tap detection accuracy of 99.3%. The mean absolute error over all trials was 7.2 mm, with a 95<sup>th</sup> percentile absolute error of 15 mm. The error disproportionately affected points far from the sensor array. Figure 6 shows a plot of mean displacement from the requested point (black line) and the standard deviation (red ellipse) for each point and projection location.

### Swipe Accuracy Results

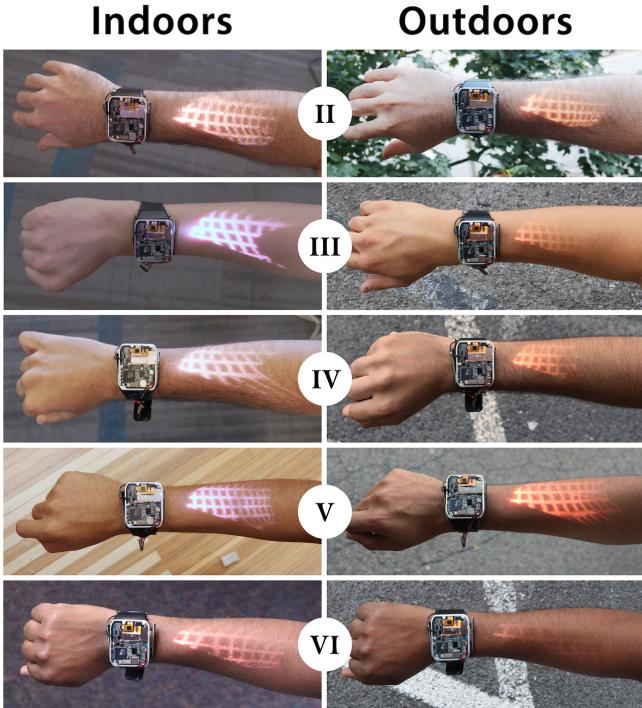
For swipes, the detection accuracy was 100%, with an average swipe-direction classification accuracy of 96.0%, (detailed breakdown offered in Table 1). The biggest source of error was down-swipes being misclassified as up-swipes. From observing participants during the study, we suspect this is due to users navigating their finger to the start of a down-sweep too close to the skin, causing the pre-gesture motion to be classified as a swipe itself, leading to an incorrect classification.

		Classified Gesture (%)			
		Left	Right	Up	Down
Actual Gesture (%)	Left	99.3	0.7	0.0	0.0
	Right	0.0	99.3	0.7	0.0
Up	1.3	1.3	96.7	0.7	
Down	2.0	2.0	7.3	88.7	

**Table 1. Swipe classification confusion matrix**

### Study 2: Projection Performance

Skin color significantly affects projection visibility due to variable absorption of light. In our second study, we wanted to evaluate projection performance across skin tones and lighting conditions. We recruited five participants, one for each skin type from II through VI on the Fitzpatrick scale [9], with varying levels of hair (Figure 7). Each participant was asked to wear the watch on their right arm for consistency (projecting onto the top of the arm), and then swipe the unlock interface to present a rectified 10 mm grid. Notably, the arm calibration used for this study was the same for all participants, *i.e.* no per-user arm model was em-



**Figure 7. Results from the projection performance study.**  
Roman numerals indicate Fitzpatrick skin type.

ployed. To evaluate the projection quality under different lighting conditions, this process was performed once indoors under typical office fluorescent lighting, and once again outdoors on a sunny day.

#### Visibility

Photos of each participant and condition are shown in Figure 7. Projections were clearly visible indoors on all skin types, and projections were generally visible outdoors (with the exception of skin type VI). All projected interfaces were hard to see in direct sunlight (also true of phone screens).

#### Rectification Accuracy

To evaluate the accuracy of our rectified graphics, we took photographs and *post hoc* fit lines to the projected grid. We first computed the angle of each line relative to the watch, comparing it against the intended axes. On average, grid lines deviated from the ideal by a mean absolute error of 10.7° (SD=8.0), with a 95<sup>th</sup> percentile deviation of 29.0°. For axially-aligned grid lines, the mean absolute deviation was 3.7° (SD=3.2) with a 95<sup>th</sup> percentile deviation of 8.4°. Several of our grids show a slightly “skewed” projection (*e.g.*, Figure 7, III indoors) due to twisting of the watch face, resulting in higher error for tangential lines.

#### Scale Accuracy

To assess scale accuracy, we measured the size of each projected grid square, and compared it with the intended size of 10×10 mm. In the axial direction, the mean absolute error was 22.0% (SD=19.4), with a 95<sup>th</sup> percentile error of 57%. In the tangential direction, the mean absolute deviation was 13% (SD=9.4), with a 95<sup>th</sup> percentile error of 33%.

## INTERACTIONS

We developed a simple APIs that allows developers to treat projected arms as conventional touchscreens. Developers author their interfaces using a fixed DPI (2 pixels per mm in our current implementation), and the resulting imagery is transferred to a texture and rendered onto the arm in a rectified manner. Due to varying arm angles, the available surface area might change from session to session; our API can provide the extent of the projected area so that applications can adjust their appearance accordingly. Touch and swipe events are also provided by the API via callbacks, and applications can additionally access the raw finger position for more advanced sensing purposes. These APIs enable our LumiWatch to readily support standard single-finger touch interactions on the arm.

#### Finger Taps

Our sensor array has a limited vertical field of view due to its orientation, causing it to only register fingers that are within ~1 cm of the skin surface. Thus, touching a finger to the skin and then lifting it more than ~1 cm off the surface is considered a tap-and-release gesture (*i.e.*, a single finger tap). For example, a user can use this tap (“click”) gesture to select and launch an application from the watch’s home screen (Figure 1B).

#### Continuous Finger Tracking

Our sensor array provides continuous tracking of an in-range finger. Thus, the user can provide continuous positional input by moving their fingertip across the skin. For instance, users could use to draw (Figure 4), perform stroked input (*e.g.* Graffiti [27] or Unistroke Gestures [12] for text input), or adjust a continuous variable (*e.g.*, slider or knob; Figure 1C). Continuous input also lends itself to common, single-touch motion gestures, such as panning a map, scrolling a list, or swiping to select/dismiss an item (Figure 1D, see also Video Figure).

## DISCUSSION & LIMITATIONS

#### Scanned-Laser Projector

The primary advantage of our scanned laser design, over similar small-form-factor designs such as LCoS or DLP, is that the light source is fully redirected towards projection. In LCoS or DLP systems, a large percentage of the light source is redirected or absorbed to produce black or dark imagery. By using a scanned laser, we save power and simultaneously enable brighter images for a fraction of the lumens – our internal tests suggest that our projector is equivalent to about 100 lumens of DLP projection for typical content (*e.g.*, videos). Scanned-laser designs are also focus-free (*i.e.*, in focus at all distances), which simplifies the optics, allowing our design to be compact.

However, a major drawback of scanned lasers is laser safety. Large projectors do not use scanned lasers because the laser power would exceed safe limits in the event of direct retinal contact. A commonly cited analysis of picoprojector laser safety [4] finds 17 lumens to be the upper bound for class 2 safety. However, a later analysis [10] suggests that

existing pulsed-laser analyses are too conservative for scanned laser projectors, and proposes that projectors up to 39 lumens could be safely classed as class 2. The IEC, which oversees laser safety standards, has yet to offer guidance specific to scanned laser projectors, and so we use a more conservative value of 15 lumens in our prototype. In the future, as the laser safety issues are better understood and regulated, higher-power designs may appear, which will provide better usability in a wider array of conditions.

### Heat Dissipation

The inherent small size of smartwatches limits their heat dissipation capability, which causes our watch to heat up considerably during projection. To avoid damaging the components, we set a strict 65°C limit on internal component temperatures before shutting down the projector. In practice, this limit can be reached within minutes if a full-brightness white image is displayed continuously.

Vents and fans are a common solution to this problem, but cumbersome in a small and energy-limited form factor. There are a few potential strategies to improve heat dissipation. First, our current design dissipates very little heat at the watch-skin interface, as we placed the battery at the bottom of the watch body. A future design could incorporate a metallic case thermally coupled to the logic board and projector, which could dissipate some heat to the wearer. A second, more radical possibility is to redesign the watch as a wristband, with hot components better distributed, and also using the watch and straps as heat sinks. This would provide a larger surface area for heat dissipation (both to skin and air), though it increases manufacturing complexity.

### Irregular Usable Area

Our projected area has an irregular shape, which makes developing applications more challenging than traditional rectilinear screens. Moreover, the farthest extent of the projection has limited resolution and brightness, complicating application development further. Currently, the projection resolution on the axial axis is around 17 px/mm near the wrist (30 mm from the projector), but only around 1.8 px/mm toward the elbow (130 mm from the projector). This loss of resolution implies a corresponding decrease in projection brightness. Ideally, when projecting on the arm, we would drive the mirrors at variable rates to regularize the resolution and brightness across the projection area, a challenge we leave to future work.

On a more minor note, our current design places the projector towards one corner of the watch for logic board design and heat dissipation reasons. This creates a slightly pointed projection, as opposed to trapezoidal (as would be seen if the projector was centered on the arm). Again, this introduces a layer of complexity that must be dealt with at the application layer.

### Projection Angle

Many of our early designs used a projector that was much lower to the arm's surface (~8 mm), but we found that this

produced poor results on hairy arms as a large fraction of the light was lost before illuminating the skin. Furthermore, the hairs themselves become illuminated and interfere with content rendered on skin below. This suggests that there is an ideal minimum height for the projector (roughly 10-14 mm), below which projection may simply be impractical no matter how good the projection technology is. Additionally, a shallow angle of projection also results in occlusions (shadows cast by fingers) whenever a user touches the arm, which is unavoidable in this form factor.

### Touch Sensing

Our current touch sensing approach only provides accurate finger position estimates in a rectangle bounded by our 1D sensor array's width. Outside of this region, a finger will not be in line with any sensor, and instead lies simply within its cone of sensing. This causes our algorithm to judge the finger to be further away than it actually is, reducing positional accuracy. One possible solution would be to use slightly bowed sensor array, providing not a rectangular sensing area, but a cone, better matching the geometry of the arm, though this would be harder to fabricate. A second issue is that our current approach uses ten discrete time-of-flight sensors, which are relatively large and expensive (compared to other commodity components). Therefore, we plan to explore if the number of sensors can be reduced with minimal impact to accuracy.

### CONCLUSION

In this paper, we have presented LumiWatch, a first-of-its-kind smartwatch with integrated projector and touch sensing in a commercially viable form factor. Developing this prototype required solving a number of difficult problems, including development of a suitable projector module, shallow-angle projection onto curved arms, and accurate 2D finger tracking. Through a combination of custom hardware and software, our prototype smartwatch provides a large touchscreen-like interface directly on a wearer's arm. LumiWatch presents a novel combination of hardware and software capabilities, moving the vision of on-skin interaction significantly closer to reality and illuminating the imminent feasibility of projection-enabled wearables.

### ACKNOWLEDGEMENTS

We would like to thank Hai-Wei Ding, Xin Jiang and Ming Zhang for their help in developing the hardware, and also Song Wu for his leadership.

### REFERENCES

1. Daniel L. Ashbrook, James R. Clawson, Kent Lyons, Thad E. Starner, and Nirmal Patel. 2008. Quickdraw: the impact of mobility and on-body placement on device access time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 219-222. DOI: <https://doi.org/10.1145/1357054.1357092>
2. Daniel Ashbrook, Kent Lyons, and Thad Starner. 2008. An investigation into round touchscreen wristwatch interaction. In *Proceedings of the 10th international con-*

- ference on Human computer interaction with mobile devices and services (MobileHCI '08). ACM, New York, NY, USA, 311-314.  
DOI=<http://dx.doi.org/10.1145/1409240.1409276>
3. Angela Barnett. 2009. The dancing body as a screen: Synchronizing projected motion graphics onto the human form in contemporary dance. *Comput. Entertain.* 7, 1, Article 5 (February 2009), 32 pages. DOI: <https://doi.org/10.1145/1486508.1486513>
  4. Edward Buckley. 2010. Eye-safety analysis of current laser-based scanned-beam projection systems. *Journal of the Society for Information Display*, 18: 944–951. doi:10.1889/JSID18.11.944
  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. Cicret. 2014. Cicret Bracelet: <https://cicret.com>.
  7. Nilofar Dezfuli, Mohammadreza Khalilbeigi, Jochen Huber, Florian Müller, and Max Mühlhäuser. 2012. PalmRC: imaginary palm-based remote control for eyes-free television interaction. In *Proceedings of the 10th European Conference on Interactive TV and Video (EuroITV '12)*. ACM, New York, NY, USA, 27-34. DOI: <https://doi.org/10.1145/2325616.2325623>
  8. Leo Donnelly, Debra Patten, Pamela White and Gabrielle Finn. Virtual human dissector as a learning tool for studying cross-sectional anatomy. *Med Teach.* 2009 Jun, 31(6):553-555. doi: 10.1080/01421590802512953.
  9. Thomas B. Fitzpatrick. (1975). "Soleil et peau" [Sun and skin]. *Journal de Médecine Esthétique* (2): 33–34
  10. Annette Frederiksen, Reinhold Fieß, Wilhelm Stork, Siegwart Bogatscher and Nico Heußner. 2011. Eye safety for scanning laser projection systems. *Biomed Tech (Berl)*. 2012 May 31;57(3):175-184. doi: 10.1515/bmt-2011-0088.
  11. Kate Alicia Gavaghan, Sylvain Anderegg, Matthias Peterhans, Thiago Oliveira-Santos, and Stefan Weber. 2011. Augmented reality image overlay projection for image guided open liver ablation of metastatic liver cancer. In *Proceedings of the 6th international conference on Augmented Environments for Computer-Assisted Interventions (AE-CAI'11)*. Springer-Verlag, Berlin, Heidelberg, 36-46.  
DOI=[http://dx.doi.org/10.1007/978-3-642-32630-1\\_4](http://dx.doi.org/10.1007/978-3-642-32630-1_4)
  12. David Goldberg and Cate Richardson. 1993. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 80-87.  
DOI=<http://dx.doi.org/10.1145/169059.169093>
  13. Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary phone: learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM, New York, NY, USA, 283-292. DOI: <https://doi.org/10.1145/2047196.2047233>
  14. Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*. ACM, New York, NY, USA, 441-450. DOI: <https://doi.org/10.1145/2047196.2047255>
  15. Chris Harrison, Shilpa Ramamurthy, and Scott E. Hudson. 2012. On-body interaction: armed and dangerous. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*, Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 69-76. DOI: <https://doi.org/10.1145/2148131.2148148>
  16. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 453-462. DOI: <https://doi.org/10.1145/1753326.1753394>
  17. Scott E. Hudson, Chris Harrison, Beverly L. Harrison, and Anthony LaMarca. 2010. Whack gestures: inexact and inattentive interaction with mobile devices. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction (TEI '10)*. ACM, New York, NY, USA, 109-112. DOI: <http://dx.doi.org/10.1145/1709886.1709906>
  18. Hsin-Liu (Cindy) Kao, Christian Holz, Asta Roseway, Andres Calvo, and Chris Schmandt. 2016. DuoSkin: rapidly prototyping on-skin user interfaces using skin-friendly materials. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 16-23. DOI: <https://doi.org/10.1145/2971763.2971777>
  19. Meng-Hsiung Kiang, Olav Solgaard, Kam Y. Lau and Richard S. Muller, "Electrostatic combdrive-actuated micromirrors for laser-beam scanning and positioning," in *Journal of Microelectromechanical Systems*, vol. 7, no. 1, pp. 27-37, Mar 1998. doi: 10.1109/84.661381
  20. Rebecca K. Kramer, Carmel Majidi, and Robert J. Wood. 2011. Wearable tactile keypad with stretchable artificial skin. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*. IEEE, 1103-1107.
  21. 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>
22. 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: <https://doi.org/10.1145/2984511.2984582>
  23. Rong-Hao Liang, Shu-Yang Lin, Chao-Huai Su, Kai-Yin Cheng, Bing-Yu Chen, and De-Nian Yang. 2011. SonarWatch: appropriating the forearm as a slider bar. In *SIGGRAPH Asia 2011 Emerging Technologies* (SA '11). ACM, New York, NY, USA, Article 5, 1 page. DOI: <https://doi.org/10.1145/2073370.2073374>
  24. Soo-Chul Lim, Jungsoon Shin, Seung-Chan Kim, and Joonah Park. "Expansion of Smartwatch Touch Interface from Touchscreen to Around Device Interface Using Infrared Line Image Sensors." *Sensors* 15, no. 7 (2015): 16642-16653.
  25. Shu-Yang Lin, Chao-Huai Su, Kai-Yin Cheng, Rong-Hao Liang, Tzu-Hao Kuo, and Bing-Yu Chen. 2011. Pub - point upon body: exploring eyes-free interaction and methods on an arm. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11). ACM, New York, NY, USA, 481-488. DOI: <https://doi.org/10.1145/2047196.2047259>
  26. Kent Lyons, Thad Starner, Daniel Plaisted, James Fussia, Amanda Lyons, Aaron Drew, and E. W. Looney. 2004. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '04). ACM, New York, NY, USA, 671-678. DOI: <http://dx.doi.org/10.1145/985692.985777>
  27. I. Scott MacKenzie and Shawn X. Zhang. 1997. The immediate usability of graffiti. In *Proceedings of the conference on Graphics interface '97*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 129-137.
  28. Steve Mann. 1997. "Smart clothing": wearable multi-media computing and "personal imaging" to restore the technological balance between people and their environments. In *Proceedings of the fourth ACM international conference on Multimedia* (MULTIMEDIA '96). ACM, New York, NY, USA, 163-174. DOI: <http://dx.doi.org/10.1145/244130.244184>
  29. Daniel C. McFarlane and Steven M. Wilder. 2009. Interactive dirt: increasing mobile work performance with a wearable projector-camera system. In *Proceedings of the 11th international conference on Ubiquitous computing* (UbiComp '09). ACM, New York, NY, USA, 205-214. DOI: <http://dx.doi.org/10.1145/1620545.1620577>
  30. Pranav Mistry, Pattie Maes, and Liyan Chang. 2009. WUW - wear Ur world: a wearable gestural interface. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '09). ACM, New York, NY, USA, 4111-4116. DOI: <https://doi.org/10.1145/1520340.1520626>
  31. Adiyan Mujibiya, Xiang Cao, Desney S. Tan, Dan Morris, Shwetak N. Patel, and Jun Rekimoto. 2013. The sound of touch: on-body touch and gesture sensing based on transdermal ultrasound propagation. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces* (ITS '13). ACM, New York, NY, USA, 189-198. DOI: <http://dx.doi.org/10.1145/2512349.2512821>
  32. NTT IT Corp. 2010. TenoriPop: <http://tenoripop.com>.
  33. Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: adapting skin as a soft interface. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13). ACM, New York, NY, USA, 539-544. DOI: <http://dx.doi.org/10.1145/2501988.2502039>
  34. Masa Ogata, Ryosuke Totsuka, and Michita Imai. 2015. SkinWatch: adapting skin as a gesture surface. In *SIGGRAPH Asia 2015 Emerging Technologies* (SA '15). ACM, New York, NY, USA, Article 22, 2 pages. DOI: <https://doi.org/10.1145/2818466.2818496>
  35. Debra Patten. 2007. What lies beneath: the use of three-dimensional projection in living anatomy teaching. *The Clinical Teacher*, 4: 10-14. doi:10.1111/j.1743-498X.2007.00136.x
  36. E. Rehmi Post and Margaret Orth. 1997. Smart Fabric, or "Wearable Clothing". In *Proceedings of the 1st IEEE International Symposium on Wearable Computers* (ISWC '97). IEEE Computer Society, Washington, DC, USA, 167-.
  37. Ritot. 2014. Ritot Watch: <http://ritot.com/>.
  38. T. Scott Saponas, Chris Harrison, and Hrvoje Benko. 2011. PocketTouch: through-fabric capacitive touch input. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11). ACM, New York, NY, USA, 303-308. DOI: <https://doi.org/10.1145/2047196.2047235>
  39. T. Scott Saponas, Desney S. Tan, Dan Morris, Ravin Balakrishnan, Jim Turner, and James A. Landay. 2009. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09). ACM, New York, NY, USA, 167-176. DOI: <https://doi.org/10.1145/1622176.1622208>

40. Nobuchika Sakata, Teppei Konishi, and Shogo Nishida. 2009. Mobile Interfaces Using Body Worn Projector and Camera. In *Proceedings of the 3rd International Conference on Virtual and Mixed Reality: Held as Part of HCI International 2009* (VMR '09), Randall Shumaker (Ed.). Springer-Verlag, Berlin, Heidelberg, 106-113. DOI:[http://dx.doi.org/10.1007/978-3-642-02771-0\\_12](http://dx.doi.org/10.1007/978-3-642-02771-0_12)
41. Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 483-492. DOI:  
<http://dx.doi.org/10.1145/2207676.2207743>
42. Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: projected visualizations for hand movement guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, New York, NY, USA, 179-188. DOI: <http://dx.doi.org/10.1145/2207676.2207702>
43. Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17). ACM, New York, NY, USA, 3891-3902. DOI:  
<https://doi.org/10.1145/3025453.3026005>
44. Chris Sugrue. 2007. "Delicate Boundaries"
45. Desney Tan, Dan Morris, and T. Scott Saponas. 2010. Interfaces on the go. *XRDS* 16, 4 (June 2010), 30-34. DOI: <https://doi.org/10.1145/1764848.1764856>
46. Bruce H. Thomas, Karen Grimmer, Joanne Zucco and Steve Milanese. 2002. Where Does the Mouse Go? An Investigation into the Placement of a Body-Attached TouchPad Mouse for Wearable Computers. *Personal Ubiquitous Comput.* 6, 2 (January 2002), 97-112. DOI:<http://dx.doi.org/10.1007/s007790200009>
47. Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 2991-3000. DOI:  
<https://doi.org/10.1145/2702123.2702391>
48. Andrew D. Wilson and Hrvoje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23rd 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>
49. Goshiro Yamamoto and Kosuke Sato. PALMbit: A PALM Interface with Projector-Camera System. In Adj. Proc. UbiComp '07. 276-279.
50. Cheng Zhang, AbdelKareem Bedri, Gabriel Reyes, Bailey Bercik, Omer T. Inan, Thad E. Starner, and Gregory D. Abowd. 2016. TapSkin: Recognizing On-Skin Input for Smartwatches. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces* (ISS '16). ACM, New York, NY, USA, 13-22. DOI:  
<https://doi.org/10.1145/2992154.2992187>
51. Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). ACM, New York, NY, USA, 1491-1503. DOI:  
<https://doi.org/10.1145/2858036.2858082>
52. 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>