Sridhar, Srinath; Markussen, Anders; Oulasvirta, Antti; Theobalt, Christian; Boring, Sebastian

WatchSense : On- and Above-Skin Input Sensing through a Wearable Depth Sensor

Published in:

DOI:
10.1145/3025453.3026005

Published: 01/01/2017

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor

Srinath Sridhar\textsuperscript{1}, Anders Markussen\textsuperscript{2}, Antti Oulasvirta\textsuperscript{3}, Christian Theobalt\textsuperscript{1}, Sebastian Boring\textsuperscript{2}
\textsuperscript{1}Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
\textsuperscript{2}University of Copenhagen, Denmark
\textsuperscript{3}Aalto University, Finland
\textsuperscript{1}\{ssridhar, theobalt\}@mpi-inf.mpg.de, \textsuperscript{2}\{amark, sebastian.boring\}@di.ku.dk, \textsuperscript{3}antti.oulasvirta@aalto.fi

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{(a) WatchSense enables on- and above-skin input on the back of the hand (BOH) through a wrist-worn depth sensor. (b) Our prototype mimics a smartwatch setup by attaching a small depth camera to the forearm. (c) It tracks the 3D position of fingertips, their identities, and touch on the BOH in real-time on consumer mobile devices. This enables a combination of mid-air and multitouch input for interactive applications on the move.}
\end{figure}

ABSTRACT

This paper contributes a novel sensing approach to support on- and above-skin finger input for interaction on the move. WatchSense uses a depth sensor embedded in a wearable device to expand the input space to neighboring areas of skin and the space above it. Our approach addresses challenging camera-based tracking conditions, such as oblique viewing angles and occlusions. It can accurately detect fingertips, their locations, and whether they are touching the skin or hovering above it. It extends previous work that supported either mid-air or multitouch input by simultaneously supporting both. We demonstrate feasibility with a compact, wearable prototype attached to a user’s forearm (simulating an integrated depth sensor). Our prototype—which runs in real-time on consumer mobile devices—enables a 3D input space on the back of the hand. We evaluated the accuracy and robustness of the approach in a user study. We also show how WatchSense increases the expressiveness of input by interweaving mid-air and multitouch for several interactive applications.

Author Keywords

Depth sensor; Skin interaction; Smartwatch; Finger tracking.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces – Input Devices and Strategies; I.4.8 [Image processing and computer vision]: Scene Analysis – Tracking

INTRODUCTION

This paper studies novel input capabilities enabled by computer vision sensing on small wearable devices such as smartwatches. Every new generation of such devices features better displays, processors, cameras, and other sensors. However, their small form factor imposes severe limitations on the efficiency and expressiveness of input. Touch input is restricted to a tiny surface, and gesture input may require moving a whole body part [2]. We address these challenges by investigating a class of emerging sensing techniques that support extending the input space to the space next to a wearable device. This could solve the problems caused by small interactive surface area. Additionally, it may enable a new possibility for multi-device interaction: controlling not only the wearable device itself but also relaying sensed input to allow interaction with nearby devices, such as TVs, smartphones, and virtual/augmented reality (VR/AR) glasses [16, 24].

We contribute to an emerging line of research exploring richer use of finger input sensed through a wearable device. In particular, we look at smartwatches, which have previously been
supplemented by mid-air finger input [14, 20, 21, 22]. Recent papers propose using the palm or forearm for gestures or touch input (e.g., [25, 33, 34, 37]). This enlarges the size of input space in which gestures can be comfortably performed. However, previous papers focused on either touch or mid-air interactions. We address the combination of these two modalities, with the aim of increasing the efficiency and expressiveness of input. Recent advances in depth sensor miniaturization have led to the exploration of using both touch and mid-air interactions above smartphones [7]. To our knowledge, there is no work that explores the use of both touch and mid-air input in smaller, wearable form factor devices such as smartwatches.

Our second contribution is to address the technical challenges that arise from sensing of fingers that touch the skin and/or hover above the skin near a smartwatch with an embedded depth sensor. Recent improvements to real-time finger tracking in mid-air [19, 28, 31, 38] cannot directly be employed due the oblique camera view and resulting occlusions which are common in body-worn cameras. To address these challenges we propose a novel algorithm that combines machine learning, image processing, and robust estimators.

Our novel method allows joint estimation of 3D fingertip positions, detection of finger identities, and detection of fingertips touching the back of the hand (BOH). Unlike previous work [13, 22], our approach also detects finger identities which can further increase input expressiveness. Our prototype (Figure 1 (b, c)), which mimics the viewpoint of future embedded depth sensors, can detect fingertips, their identities, and touch events in real-time (> 250 Hz on a laptop and 40 Hz on a smartphone). Additionally, technical evaluations show that our approach is accurate and robust for users with varying hand dimensions.

The capability enabled by our approach allows for simultaneous touch and mid-air input using multiple fingers on and above the BOH. A unified sensing approach that supports both touch and mid-air, as well as finger identity detection is not only beneficial for users but also provides more interaction design possibilities. We show through several applications that this novel input space (or volume) can be used for interaction on the move (e.g., with the smartwatch itself or with other nearby devices), complementing solutions with touch or mid-air alone. In summary, our paper contributes by:

- Exploring the interaction space of on- and above-skin input near wearable devices, particularly smartwatches.
- Addressing the technical challenges that make camera-based sensing of finger positions, their identities, and on-skin touch a hard problem.
- Demonstrating the feasibility of our approach using a prototype, technical evaluations, and interactive applications.

**WATCHSENSE**

Figure 1 (a) illustrates the vision of WatchSense. We assume that smartwatches will embed a depth sensor on their side, overseeing the back of the hand (BOH) and the space above it. In this section, we first outline the vision of embedded depth sensors and how we prototype this vision. Then, we outline the new interaction opportunities afforded by WatchSense, and present the arising tracking challenges.

**Embedded Depth Sensors**

Advances in time of flight (TOF) imaging technology have led to rapid miniaturization of depth cameras. A few years ago, the smallest TOF sensor (Swissranger SR4000) had a size of $65 \times 65 \times 68$ mm. Today, the PMD CamBoard Picoflexx measures only $68 \times 17 \times 7.25$ mm. While these sensors do not yet fit into a smartwatch, the trend indicates that smaller sensors will be integrated into smartwatches in the near future.

To study the utility of such embedded sensors already, we created a prototype with viewing angles close to a hypothesized integrated depth sensor. Figure 1 (b) shows our prototype setup: a small depth sensor is attached to the user’s forearm facing the wrist. Due to near range sensing limitations of these sensors (usually designed for sensing up to 2 m) we had to place them at a distance of 20 cm from the wrist. However, we envision specially designed future TOF sensors will allow better near range sensing capabilities.

**Input Capabilities**

WatchSense is capable of sensing fingertip positions and identities (on the interacting hand) on and above the BOH. This opens up new interaction opportunities for multi-finger interactions—both while touching the BOH as well as in mid-air. The resulting input space provides higher expressiveness and degrees of freedom than skin-based touch. While this is interesting for input directly to smartwatches, we envision that watches will become an input sensing device for a large variety of other interactive devices (see WATCHSENSE-ENABLED APPLICATIONS section for examples). Figure 2 highlights the possible interaction combinations with WatchSense.

**Touch and Mid-Air Tracking:** With WatchSense, the BOH can be used as a touchpad with the same operations: sensing when a touch operation began, when the finger moved (reporting its $x, y$ coordinates in the plane, where $z$ is 0), and when it is lifted (see Figure 2 (a)). Additionally, sensing the space above the BOH allows for using mid-air gestures (see Figure 2 (b)). Here, however, the sensor reports 3D $x, y, z$ coordinates. Thus, WatchSense offers 3 degrees of freedom (DoF) per finger. Transitioning between touch and mid-air input allows for similar interactions as shown in Air+Touch [7].

**Finger Identification:** WatchSense supports the identification of fingers (see Figure 2 (c)). For instance, this allows for assigning different interactions to different fingers (i.e., touching or gesturing with the thumb has a different meaning than when doing so with the index finger). While we envision identifying all five fingers, in this paper we showcase the interaction opportunities with the thumb and index finger.

**Multi-Finger Touch & Mid-Air:** Combining finger identification with touch and mid-air sensing (and the resulting 3 DoFs per finger) enables compound interactions. The matrix in Figure 2 (d) showcases the possible combinations, and the examples presented later in this paper highlight their use.

---

1Swissranger SR4000: http://hptg.com/industrial/

2CamBoard Picoflexx: http://pmdtec.com/picoflexx/
When the interacting hand is present, each finger is either touching the BOH, or positioned mid-air. We use the following terminology throughout the paper: the overall interaction state is described by a tuple containing the thumb’s state and the index’s state (i.e., if the thumb is touching, and the index is not, the overall state is \textit{Touch + Mid-Air}).

These combinations can be used with large variation. For example, in \textit{Touch + Mid-air}, the hand can be utilized as a joystick, where the thumb acts as base, while the index finger rotates around that base. In \textit{Touch + Touch}, the BOH can be utilized as a multitouch surface. \textit{Mid-air + Touch} is can be utilized when using the BOH as a single-touch touchpad. However, the thumb’s mid-air position (and distance to the index finger may be used for value-changing operations (e.g., adjusting the volume of a music player). Lastly, in \textit{Mid-air + Mid-air}, both fingers can gesture freely in 3D. We use this last state as a delimiter for entry/exit to other states.

**Resulting Challenges**

We assume that a camera obtains an oblique depth map of the BOH and the space directly above it. This differs greatly from previous approaches that use depth sensing for multitouch input. \textit{Imaginary Phone} [12] and \textit{OmniTouch} [13] assumed a near-perpendicular view of the surface, easing separation of the interaction surface from the interacting hand. These systems showed limited accuracy when distinguishing touch and hover states (e.g., \textit{OmniTouch} reports 20 mm accuracy). Other systems, such as \textit{Air+Touch} [7] rely on a perfectly planar, touch-sensitive surface on a smartphone in addition to the depth sensor.

Realizing our scenario without additional sensors on the hand poses new challenges: (1) the oblique view of the BOH causes perspective distortion and additional occlusions, (2) the BOH (as well as the forearm) is not a flat surface but curved, which complicates touch detection, (3) multi-finger interaction requires the discrimination and identification of fingertips, both when touching and hovering, and (4) compute limitations on mobile devices require the sensing technique to be fast with low latency. \textit{WatchSense} supports simultaneous and continuous touch and mid-air interactions from an oblique view of the BOH in real-time—even in the presence of these challenges.

**RELATED WORK**

The work presented in this paper builds on recent advances in interacting on smartwatches and on associated limbs, mid-air interaction techniques around wearable devices, as well as hand and finger tracking.

**Touch Interaction On and Around Smartwatches**: Interaction with consumer smartwatches is generally limited to touch and speech. Two main strategies have been explored to extend the capabilities of such devices: (1) on-device interaction, and (2) on-body interaction.

On-device interactions beyond the touchscreen employ other parts of the smartwatch. Pasquero \textit{et al.} [26] extended input to the device’s bezel. Xiao \textit{et al.} [39] use the entire watch face for additional input, e.g., through tilting, twisting or panning it. \textit{WatchIt} uses the wristband as alternative input canvas for simple gestures [27]. \textit{WatchMI} [40] uses existing sensors to support pressure touch, twisting, and panning gestures. While each of these systems is shown to be beneficial, they only consider input directly on the device.

Smartwatches have largely planar body parts in close proximity (e.g., the hand and forearm). Thus, there is a large body of research on skin-based input to free the interaction from the watch itself. \textit{iSkin} uses a thin skin overlay to detect touch and strokes [36]. \textit{SkinInput}’s bio-acoustic sensing array allows for detecting touch directly on the skin. \textit{SkinTrack} [41] uses the body as an electrical waveguide to support touch near smartwatches. Laser-based range scanners [34, 33] as well as infrared sensors placed at the device’s borders [3, 25, 32] are vision-based approaches to detect on-skin touch and gesture interaction around a device.
Most related, however, is the use of depth cameras to detect skin-based input. *Imaginary Phone* used a depth camera to detect interaction on the palm [12] to operate a mobile phone which is not in sight. *Omnitouch* used a very similar setup to turn arbitrary (planar) surfaces (also the user’s palm or wrist) into projected, interactive surfaces [13]. *WatchSense* is inspired by these systems, but we go beyond by recognizing fingertip positions, identities, and touch on- and above-skin.

**Gestural Interaction Around Wearable Devices:** Mid-air space around wearable devices has also been investigated for input. Initially, researchers used that space for simple gestural input. *Gesture Watch* [20], *AirTouch* [22], and *HoverFlow* [21] used an array of infrared sensors to execute simple commands through eyes-free gestures. More recently, researchers began exploring techniques that rely on more accurate mid-air tracking. Here, they relied on magnetic tracking (e.g., *FingerPad* [5], *Abracadabra* [14], and *uTrack* [6]), or small infrared cameras (e.g., *Imaginary Interfaces* [11]). To test a set of interaction techniques, researchers often relied on sophisticated external tracking systems (e.g., [17, 15]). Finally, there is research on using the fingers for gestural input, either using a vision-based approach [19, 35], or through strain sensors on the back of the hand [23].

The aforementioned systems solely used gestural input without considering touch, which is a key feature of *WatchSense*. One of the few systems considering both touch and mid-air during an interaction is *Air+Touch* [7]. Their focus is on sequential interactions near smartwatches, where mid-air interaction occurs before, after or in between touches. In contrast, *WatchSense* allows for simultaneous use of touch and mid-air.

**Vision-based Tracking of Hands and Fingers:** With the advent of commodity depth sensors, research on articulated hand tracking (e.g., *Digits* [19]) has gained more attention [18, 28, 31]. These approaches aim at reconstructing hand pose from depth data, and would be, at first glance, an ideal solution for our scenario. Unfortunately, these methods fail under oblique views, occlusions, or additional objects in the scene. In addition, they are not well-suited for detecting (multi-)touch events. To bypass these issues, existing systems (that make use of finger input) often simplify the problem: first, systems avoid fully articulated hand tracking and only require detecting discrete touch points (e.g., [38, 3, 1, 7]). Second, several systems build on heuristic assumptions of the depth camera’s location in relation to the interaction surface which is hard to realize in practice. For example, both *Omnitouch* [13] and *Imaginary Phone* [12] assume a perpendicular view of the interaction surface, easing separation of the interaction surface from the interacting hand. In addition, these systems have limited accuracy when distinguishing touch and hover states (e.g., *Omnitouch* reports 20 mm accuracy [13]). Other systems, such as *Air+Touch* rely on a perfectly plain, touch-sensitive interaction surface (a smartphone) [7].

In comparison, our work builds on less heuristic assumptions while accurately detecting fingertips on and above the interacting surface. Taking inspiration from [30, 13, 22] we use a combination of machine learning, image processing, and robust estimators to solve the challenging vision problem. Our approach is flexible and can be retrained to fit a wide range of depth sensor positions (e.g., in the device itself), surfaces (e.g., upper arm). Additionally, we obtain information about finger identity that increases the expressiveness of interactions possible with our approach.

**IMPLEMENTATION**

We now describe our depth camera-based method for supporting expressive mid-air and multitouch interactions. Our focus is on fingers interacting on and above the BOH from an arm-worn camera. Our approach is fast and accurate—we can track the position of fingertips to within 15 mm, and touch points to within 10 mm. Our approach is also flexible—it can be reused with only a few changes to suit other wearable cameras, and viewpoints.

Previous methods [13, 22] for near-surface finger interaction support estimation of the following: (1) 3D hover/touch positions of fingertips, and (2) exact detection of finger touch events. Our approach supports these and additionally also (3) automatically, and robustly identifies fingertips (currently index finger and thumb). This allows us to support a richer set of mid-air and multitouch interactions. Our approach also delivers better touch detection tolerances than previous work.

**Prototype System**

Our prototype can run on desktops, laptops, tablets, and smartphones and relays sensed fingertip positions, labels, and touch events through a WebSocket connection. Clients such as smartphones, smartphones, public displays, or smartglasses can obtain this information wirelessly.

In our prototype, we use the PMD CamBoard PicoFlexx camera, which is currently the smallest commercially available depth sensor. We found its size, resolution, and noise characteristics suitable for the BOH scenario. However, we also support other close range sensors like the Intel Senz3D depth, and the Intel RealSense F200. We position the sensor on the forearm (20 cm above the wrist) facing the BOH (see Figure 1). Placing the sensor closer to the wrist was not possible because commercial TOF cameras have limited near range sensing capability. Their infrared illumination source—designed for ranges >50 cm—saturates pixels with depth less than 20 cm thus making depth estimation unreliable. Specially designed cameras with less intense illumination sources will allow nearer sensing ranges.

**Algorithm Description**

Estimating fingertip positions, and touch events from an oblique view of the BOH is a hard problem. Even state-of-the-art articulated RGB-D hand trackers would fail under these conditions [28, 31]. We use a detection rather than tracking strategy to help recover in case of failure. Our approach features a novel combination of random forests, advanced image processing, and robust estimators to achieve stable and accurate fingertip, finger identity, and touch detection. Figure 3 provides an overview of our approach.

**Random Forests for Classification:** We use per-pixel classification forests which have been shown to produce state-of-the-art results in human pose estimation and other segmentation
problems [29, 18, 31]. We only provide a brief overview (for details, see [9]). Our contribution shows that random forests in combination with other techniques enable new interaction opportunities for on- and above-skin wearable interaction.

Given an image, a classification forest is trained to label each pixel into a class label (e.g., part of a human body). At test time, for each input pixel, a tree in the forest makes a prediction about which part it likely belongs to. The output from all trees in the forest is aggregated to provide a final prediction about the pixel’s class as \( p(c | x, \tau) = \frac{1}{T} \sum_{t=1}^{T} p_t(c | x, \tau) \), where \( p \) is the predicted class distribution for the pixel \( x \) given forest hyperparameters \( \tau \). \( T \) is the number of random trees that makes a prediction \( p_t \). We use depth-based feature response functions similar to the one described in [29].

**Input Preprocessing and Segmentation:** The input depth map encodes real-world depth at each pixel. Noise in the depth map is removed using morphological erosion and a median filter to produce a filtered depth map [10]. To make subsequent steps in our method more robust, we first use a binary classification forest that segments the two interacting hands into BOH and interacting hand (see Figure 3). At training time, we composite images of the BOH only and other hand images. This allows fast generation of large amounts of annotated data. At testing time, segmentation generates two depth maps—one contains only the BOH and the other contains only the interacting hand.

**Fingertip Detection and Recognition:** The goal of this part is to detect and estimate the 3D position and identity of interacting fingertips. In our prototype, we assume that only two fingers interact (i.e., index finger and thumb)—however our approach is flexible and can support more than two fingertips. Additionally, we trained our method to be robust to false positives on unsupported fingers. The key improvement over previous work is our ability to detect fingertips and also their unique identity even after periods of occlusion. In contrast, [22] uses only one finger while [13] uses heuristics to assign unique IDs without actually knowing finger identity. As we show in the applications section, fingertip identity allows us to create more expressive interactions previously not possible.

We rely on a random forest that classifies pixels into one of three classes: IndexFinger, Thumb, Background. More classes can be added if needed. At training time, we collected color and depth image pairs from multiple users interacting with the BOH wearing colored markers (see Figure 4). These markers were automatically detected in the color image and mapped onto the depth image. This provides labels for the forest to be trained on—we collected 20000 image pairs from different users to maximize forest generalization.

At testing time, given an input depth image, the forest classifies pixels into one of the three classes. The result, shown in Figure 4, produces a group of pixels that are labelled into one of the fingertips. We remove noise in the resulting pixels by a median filter and morphological erosion. We then obtain a robust estimate for the 2D fingertip position on the image by applying the MeanShift algorithm [8] which is robust to outliers. The final 2D position is then backprojected using the depth map to obtain the 3D fingertip position along with its identity (see Figure 4).

Our approach is resilient to temporary tracking failures since the fingertips are detected frame-by-frame. For added stability, we filter the final positions with the 1σ filter [4]. Because we identify fingertips uniquely we can support more expressive interactions previously not possible, as we show in our interactive applications.

**Touch Detection:** The second goal is to robustly detect fingertips touching the BOH. This is a hard because depth sensors have noise and limited precision. The oblique camera view, general BOH shape, and camera motion make it even harder. We experimented with various techniques including distance computation from a plane fitted to the BOH. However, we found that flood filling, similar to the approach used by OmniTouch [13], worked best.

Figure 5 illustrates touch detection with flood filling. For each detected fingertip, we seed the flood filling process at its 2D position. We then fill a fixed mask around the fingertip such that pixels of a certain depth in front of and behind the fingertip (i.e., towards or away on the camera z-axis) are filled. We empirically chose the near and far thresholds to be 50 mm and 20 mm, respectively, which we found to cover a wide range of motion of the BOH and finger orientations. Whenever more than 40% of the mask is filled, we activate a touch event. For robustness, we only do so when more than 10 frames (at the device runtime framerate) in sequence were detected as...
touching. As we show later, this method’s touch detection tolerance varied from 1 mm to about 10 mm for different users which is better than the 20 mm reported by [13].

Figure 5. Touch detection. (a) When there is no touch, flood fill is restricted to filling only in parts of the finger. (b, c) When the finger touches flood fill grows into the BOH filling a larger area (White: seed point, Brown: flood filled pixels).

TECHNICAL EVALUATION

We evaluated several key performance aspects of our method: (1) accuracy of fingertip tracking while touching the BOH and hovering above it, (2) reliable minimum distances (tolerance) between finger and the BOH to separate touch and hover, and (3) classification accuracy of the random forest. We first report our method’s runtime performance.

Runtime Performance

Our approach runs in real-time on an Intel Core i7 laptop at >250 Hz, at >40 Hz on a recent smartphone (OnePlus 3), and at 35 Hz on a tablet (Nexus 9). All components of our method run completely on the CPU. Given the simplicity of our method and the increasing compute capability of wearables, we expect to be able to run our method directly on smartwatches in the future. Please see the supplementary video for a demonstration of our method running in real-time on different devices.

Touch Accuracy

The goal of this evaluation is to assess the accuracy of fingertip position and touch detection. We model our evaluation on OmniTouch [13] and SkinTrack [41].

Method: We recruited 13 right-handed volunteers (2 female) from our institution, ranging in age from 23 to 36 years (mean 28.1 years). Their backs of the hand widths varied from 70 mm to 90 mm, and lengths varied from 60 mm to 80 mm (mean dimension was 82×70 mm). The length of index fingers ranged from 69 mm to 86 mm (M=79 mm), and the thumb length was between 55 mm and 70 mm (M= 63.5 mm). Since skin color affects depth and noise at each pixel, we recruited participants with diverse skin colors. An evaluation session took around 15 minutes. Data from one participant had to be excluded because of a software issue that affected the camera.

Design and Task: The touch accuracy task measures how accurately we can detect touch points on the BOH. We had two conditions in this task: (a) in the seated condition, participants were seated and their forearm was supported by the desk, (b) in the standing condition, participants stood without any arm-support. Participants then had to repeatedly touch dots on the back of their hand using either the thumb or their index finger. The computer next to the participants showed the dot they had to touch. The experiment began when participants pressed the spacebar, which would cause the first dot to be highlighted. Then participants had to touch that dot on the back of their hand, and subsequently press the spacebar to switch to the next trial. If there was no touch recorded prior to pressing the space-bar, participants could not advance to the next trial, and an error was recorded. We recorded x, y, z-coordinates for both fingers, and which finger was touching (or not).

Apparatus: In the seated condition, participants rested their arm on a desk. The desk and chair used in our experiment were height-adjustable. The setup was replicated at two locations. Both seated and standing conditions took place in the front of a 55” 4K display or a 25” full HD display. The display and tracker were run on an Intel Xeon Dual Core (2.5 GHz) or on an Intel Xeon E3-1246 (3.5 GHz) machine. Half the participants were assigned to use the Creative Senz3D depth sensor while the other half used the PMD CamBoard PicoFlexx.

Procedure: In each of the two stages, participants either began with the index finger or the thumb, and performed all trials with that finger, before changing to the other finger. Half of our participants started with the index finger (the other half started with the thumb). The presentation of order in which the nine dots had to be touched was randomized for all tasks. In both touch accuracy tasks, each dot was touched 6 times per finger, resulting in 2 (Tasks) × 2 (Fingers) × 9 (Dots) × 6 (Repetitions) = 216 data points.

Before the experiment began, participants filled a questionnaire containing demographic information. We then measured the size of their hands as well as the length of their thumbs and index fingers. Afterwards, we fitted the prototype on the forearm, and added 3×3 dots on a participant’s back of the hand using a stencil to ensure equal separation of those dots (dots were separated by 20 mm).

Results: Figure 6 plots the distribution of touch points on the BOH—separately for standing/sitting, and the two cameras used. Black crosses represent ground truth positions. The plots show that accuracy for index finger touch positions is high in sitting and standing conditions across both two cameras. For the Senz3D, the mean standard deviation for the index finger was 4.1 mm for sitting (3.7 mm for standing). For the PicoFlexx sensor, the mean standard deviation was 5.2 mm for sitting (3.7 mm for standing). The thumb performed slightly worse for both cameras. For the Senz3D, the thumb’s mean standard deviation was 7.7 mm for sitting (8.4 mm for standing). For the PicoFlexx sensor, the mean standard deviation was 6.0 mm for sitting (7.6 mm for standing). We attribute this difference to the lack of sufficient samples for the thumb during random forest training. We, however, observe that the PicoFlexx camera performed better for the thumb than Senz3D. We would also like to highlight that our standard deviations improve over previous work [13] in spite of a smaller inter-dot distance of 20 mm instead of 30 mm.

Touch Tolerance

The goal was to assess the hover interval, where touch and hover detection is ambiguous. As we had no automated way of obtaining ground truth information for hover states, the evaluation was conducted through a series of manual measurements.
Participants: We recruited two right-handed volunteers (62 and 66 years). An evaluation session took 30 minutes.

Design, Task, and Procedure: In order to provide as reliable measurements as possible, two tables were used to support the participant’s arms during the evaluation. Participants were seated, resting their arm on one table, the other arm was resting on an adjacent elevation table with the hand hanging over the edge of the table. Before starting the evaluation, the participant’s hand was annotated with 9 dots in the same way as in the touch accuracy evaluation.

The measurements were recorded through a five step procedure: (1) The elevation table was lowered until the finger touched the BOH, (2) the BOH and finger were aligned to touch a particular dot, (3) the table was elevated to a non-ambiguous hover state, (4) the finger was then lowered in small steps (<1 mm) through the area of ambiguity and stopped when a touch state was obtained for more than 2 seconds, and (5) the finger was then elevated in similar steps until a hover state was obtained for more than 2 seconds. Measurements were recorded at the end of steps (4) and (5). The procedure was repeated for all of the nine dots for both fingers leading to 72 total dots.

Results: All measurements of non-ambiguous touch and hover states fell within an interval between 1 mm and 10 mm. This indicates that our algorithm is capable of reliably detecting a touch state at 1 mm distance from the BOH. Further, it reliably detects hovering when the finger is 10 mm away from the surface. Compared to previous state of the art [13], which reported their interval to be between 10 mm and 20 mm, this is a notable improvement.

Random Forest Classification Accuracy
Additionally, we also report accuracy of using random forests for classification. When training our classification forests, we adopted a rigorous cross-validation procedure to tune the parameters. For the best parameters chosen the per-pixel classification accuracy was 77% for fingertip detection, and 98.8% for hand segmentation.

WATCHSENSE-ENABLED APPLICATIONS
To illustrate the novel input capabilities enabled by WatchSense, we built several demonstrator applications. We thus explore how WatchSense can support existing multitouch interactions such as pinch-to-zoom, as well as open up completely new opportunities, thanks to finger identification and compound interactions. To further showcase flexibility, we show our prototype running on different hardware platforms and depth sensors. We also show cross-platform interaction, i.e., WatchSense can run on a mobile device but be used for interaction with another surrounding device (e.g., HoloLens), or even several surrounding devices simultaneously.

Music Controller: When the WatchSense app runs on a mobile device (running Android), it provides a music player con-

Figure 6. Evaluation of touch accuracy on the BOH. Each image represents the 2D touch position distribution for a particular finger, condition, and camera [Senz3D is (a)–(d), PicoFlexx is (e)–(h)]. The plots contain all touch points recorded by the tracker during each trial. Ground truth positions are marked with a black plus symbol, and ellipses denote 95% confidence intervals. Index finger performed best for both sitting and standing conditions for all cameras. We attribute the relatively worse performance of the thumb to the lack of sufficient training data for the fingertip classification forest.
We performed a pilot study to compare WatchSense with a Virtual/Augmented Reality (VR/AR) Input: We built Android controls while they took only 31.4 s with the WatchSense when users brought their hand back to the original position.

The default design used a rocker switch for volume control. Moreover, WatchSense allowed operating possible music control actions (volume up, volume down, next track, previous track, pause/play). There were two conditions: (1) default Android controls, (2) WatchSense controls. WatchSense controls were faster in particular for volume control. On average, users took 31.9 s to complete actions with the Android controls while they took only 31.4 s with the WatchSense controls. WatchSense controls were faster in particular for volume control. Moreover, WatchSense allowed operating the phone application in eyes-free fashion.

Virtual/Augmented Reality (VR/AR) Input: We built a game for virtual or augmented reality glasses called CardboardBoxes. Users can play with tens of cardboard boxes strewn across a virtual or real environment. This game showcases the 3D interaction capabilities of WatchSense. Users can select a box from the scene by gazing at an object and touching the BOH with their index finger (Mid-air + Touch, see Figure 7). Once selected, boxes can be moved around the scene or scaled. Moving is achieved by a Touch + Mid-air gesture with the index finger’s 3D position relative to the thumb being used for mapping the box’s 3D position relative to the observer. Scaling the box can be achieved by pinching on the BOH with both fingers (Touch + Touch).

In VR mode, tracking runs on a smartphone as a background app. We use the Google Cardboard API to render the game on the same device. Please see the supplementary video for more details. In AR mode, tracking can run on any device which relays sensed input to a heads-up display (we run it on a smartphone for mobility). The game is rendered on a HoloLens which allows for natural interaction with both the game and the environment. The default HoloLens interaction modality of using free hand gestures to move objects can be fatiguing for users. In contrast, our approach allows users to move objects with only finger movements. Additionally, with a pinch gesture on the BOH users can scale objects—this is not an easy task with current freehand gestures.

Map on a Watch: By combining on-BOH and mid-air input, we created more expressive interactions for a map application than a smartwatch allows. Our solution uses three interactions: (1) touching the BOH using the index finger (Mid-air + Touch), allowing for single-touch interactions like on the screen (i.e., dragging the map), (2) when both fingers touch the back of the hand (Touch + Touch), users zoom in or out, (3) when only the thumb is touching (Touch + Mid-air), a pop-up menu is shown allowing for switching display modes (map, transit, and satellite). Selection is performed by changing the distance between the two fingers through moving the index finger.

CardboardBoxes game for VR/AR. (a, b) Users select a box with the index finger (Mid-air + Touch). (c) Pinching with both fingers scales the box (Touch + Touch). (d) Thumb touching while index finger in mid-air allows translating the selected box (Touch + Mid-air).

Figure 7. CardboardBoxes game for VR/AR. (a, b) Users select a box with the index finger (Mid-air + Touch). (c) Pinching with both fingers scales the box (Touch + Touch). (d) Thumb touching while index finger in mid-air allows translating the selected box (Touch + Mid-air).

WatchSense setup to control a maps application on a smartwatch. (b) The index finger touching the BOH can be used to pan the map (Mid-air + Touch). (c) Both fingers touching while pinching zooms the map (Touch + Touch). (d) Thumb touching while index finger in mid-air allows map layer selection (Touch + Mid-air).

Figure 8. (a) WatchSense setup to control a maps application on a smartwatch. (b) The index finger touching the BOH can be used to pan the map (Mid-air + Touch). (c) Both fingers touching while pinching zooms the map (Touch + Touch). (d) Thumb touching while index finger in mid-air allows map layer selection (Touch + Mid-air).
Again, this solution is more expressive than what would be possible with touch or mid-air alone.

Controlling a Game: WatchSense also enables joystick-like input for a wearable device. This is achieved by touching the BOH with thumb and controlling pitch (forward/backward tilt of the hand) and roll (left/right tilt of the hand). Figure 10 shows this Touch + Mid-air gesture: the interacting hand forms a joystick with the thumb as base and the index finger acting as the top. Our game is a space game involving space navigation and shooting other spaceships and asteroids. Three interactions were implemented: (1) pitch controls the forward and backward motion of the spacecraft (up/down on the display), (2) roll controls the left and right movement of the spacecraft, and (3) the index finger in air controls firing by quickly moving it down and up again in a trigger-like fashion.

Limitations and Future Work

WatchSense is a solution for fast and expressive multitouch and mid-air input on and above the BOH. In particular, it supports new combinations of mid-air and multitouch input leading to more expressive input capabilities. Future work will need to address some limitations of WatchSense in both the sensing and input design dimensions.

First, our prototype is bulky because the depth sensor has to be placed about 20 cm from the wrist due to near range sensing limitations. Future work should explore depth sensing technology for near range sensing. Second, we support detection of index finger and thumb but currently do not support more fingers. Tracking more fingers can potentially enrich the input capabilities of WatchSense. Third, it might be possible to extend our algorithm to support input on and above arbitrary surfaces and manipulated objects to broaden potential application scenarios.

We demonstrated the capabilities of WatchSense that can enable new forms of expressive input. Evaluating usability of our interaction techniques was beyond the scope of the paper. Future works needs to examine the usability of new gestures made possible by WatchSense. Finally, the interactions that we propose represent only a subset of the possible interaction space. More research is needed to explore how detection of fingertips, their identities, and touch can enable richer input. Future work should also look at more complex gesture patterns that can be tracked by WatchSense.

Discussion and Conclusion

This paper has contributed to methods for extending the interaction capabilities of small form factor wearable computers. WatchSense allows extending the input space on wearable devices to the back of the hand and the space above it. The BOH offers a natural and always-available surface for input, which can now be utilized in different postures, on the move, and even if the hand is carrying an object. Because WatchSense estimates fingertip locations, identities, and touch positions, it can support interactions that were previously not possible. In particular, continuous interaction and familiar multitouch gestures like pinch can be carried out on the back of the hand and combined with mid-air gestures thus increasing the expressiveness of input. Finger identification adds the possibility to trigger events and map controls to index and thumb separately.

Detecting fingertips, their identities, and touch on the BOH in real-time from the viewpoint of an embedded depth camera is a hard computer vision problem. Direct combination of previous approaches cannot be used due to large wrist motion and the oblique viewpoint. Our novel algorithm provides a fast, accurate, and robust solution for jointly sensing fingertip locations, identities, and touch on the BOH. It tackles the issues posed by oblique viewpoint, occlusions, fast motions, and fingertip ambiguity.
Finally, we have demonstrated that the BOH can not only be used to expand the input space for smartwatches, but also for relaying input to ambient devices such as public displays. This raises the exciting new possibility of rich, device-agnostic input enabled by sensing from a wearable device.

Acknowledgements: This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation programme (grant agreements 335545, 648785, and 637991). We would like to thank Sebastian Schuon.

REFERENCES


