TouchCam: Realtime Recognition of Location-Specific On-Body Gestures to Support Users with Visual Impairments

LEE STEARNS, University of Maryland, College Park
URAN OH, University of Maryland, College Park; Carnegie Mellon University
LEAH FINDLATER, University of Maryland, College Park; University of Washington
JON E. FROEHLICH, University of Maryland, College Park; University of Washington

On-body interaction, which employs the user's own body as an interactive surface, offers several advantages over existing touchscreen devices: always-available control, an expanded input space, and additional proprioceptive and tactile cues that support non-visual use. While past work has explored a variety of approaches such as wearable depth cameras, bio-acoustics, and infrared reflectance (IR) sensors, these systems do not instrument the gesturing finger, do not easily support multiple body locations, and have not been evaluated with visually impaired users (our target). In this paper, we introduce *TouchCam*, a finger wearable to support location-specific, on-body interaction. TouchCam combines data from infrared sensors, inertial measurement units, and a small camera to classify body locations and gestures using supervised learning. We empirically evaluate TouchCam's performance through a series of offline experiments followed by a realtime interactive user study with 12 blind and visually impaired participants. In our offline experiments, we achieve high accuracy (>96%) at recognizing coarse-grained touch locations (*e.g.*, palm, fingers) and location-specific gestures (*e.g.*, tap on wrist, left swipe on thigh). The follow-up user study validated our real-time system and helped reveal tradeoffs between various on-body interface designs (*e.g.*, accuracy, convenience, social acceptability). Our findings also highlight challenges to robust input sensing for visually impaired users and suggest directions for the design of future on-body interaction systems.

CCS Concepts[1]: • **Human-centered computing~Gestural input** • **Human-centered computing~Mobile devices** • *Human-centered computing~Accessibility technologies* • *Computing methodologies~Machine learning algorithms*

KEYWORDS

On-body input; Accessibility; Wearable sensors; Computer Vision Applications; Blind and Low-Vision Users; Skin Texture Classification; Gesture Recognition

ACM Reference format:

Lee Stearns, Uran Oh, Leah Findlater, and Jon E. Froehlich. 2017. TouchCam: Realtime Recognition of Location-Specific On-Body Gestures to Support Users with Visual Impairments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* #, #, Article ## (December 2017), 23 pages.
DOI: ##.####/####

1 INTRODUCTION

On-body interaction, which employs the user's own body as an interactive surface, is an emerging area of research that offers several advantages over existing touchscreen devices, particularly for non-visual use (*e.g.*,

This work is supported by the Office of the Assistant Secretary of Defense for Health Affairs under Award W81XWH-14-1-0617.
Contact author's address: L. Stearns, Department of Computer Science, 3173 AV Williams, University of Maryland, College Park, MD 20742.

2474-9567/2017/12-ARTx $15.00
DOI: 10.1145/1234

blind users). Taps, swipes, or other on-body gestures provide lightweight and always-available control (e.g., [17,21]) with an expanded input space compared to small-screen wearable devices like smartwatches (e.g., [27,31,40,42]). In addition, the proprioceptive and tactile cues afforded by on-body input can improve eyes-free interaction (e.g., [10,32,35,57]) and enable accurate input even without visual feedback compared to the smooth surface of a touchscreen [18,45]. These advantages are particularly compelling for users with visual impairments, who do not benefit from visual cues and who frequently possess a heightened sense of tactile acuity ([15,39]).

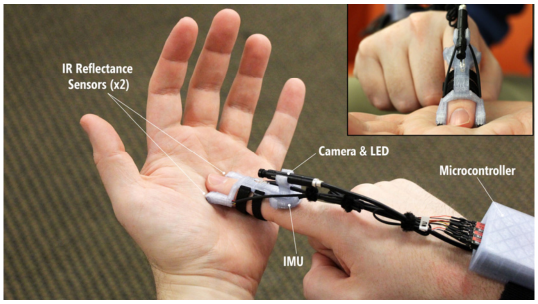


Fig. 1. *TouchCam* combines a finger-worn camera with wearable motion trackers to support location-specific, on-body interaction for users with visual impairments. See supplementary video for a demonstration.

Reliably sensing on-body input, however, is an open challenge. Researchers have explored a variety of approaches such as cameras (e.g., [4,21,50]), infrared-reflectance sensors [26,27,40], and bio-acoustics [20,28]. While promising, this prior work has not been specifically designed for or tested with visually impaired users, who likely have different needs and preferences. For example, blind users may encounter difficulty accurately aiming a camera (or other directed sensor) [25,55] and also rely more on their sense of touch [15,39] making it especially important to avoid covering the fingertips. Furthermore, prior work does not support complex gestures at multiple body locations. For example, Skinput [20] detects touches at a range of locations but not more complex gestures. In contrast, FingerPad [6] and PalmGesture [57] sense shape gestures performed on the fingertip or palm but cannot easily be extended to other locations.

Our research explores an alternative approach called *TouchCam* (Figure 1), a custom designed finger wearable to support location-specific, on-body interaction. We previously demonstrated the feasibility of recognizing body locations from small skin surface images (1–2 cm) captured using a handheld camera [53]. However, this work did not include sensor fusion, use a wearable form factor, or function in real-time. In addition, [53]'s prototype could only recognize locations (not gestures), and the images were collected under carefully controlled conditions. In this paper, we build on that work and address these limitations. TouchCam combines data from infrared reflectance (IR) sensors, inertial measurement units (IMU), and a small camera to classify body locations and gestures using supervised learning. Because TouchCam instruments the *gesturing* finger, on-body interaction is supported on a variety of locations within the user's reach while also mitigating camera framing issues. TouchCam also enables new location-specific, contextual gestures that are semantically meaningful (*e.g.*, tapping on the wrist to check the time or swiping on the thigh to control a fitness app). These features allow for flexible interface designs that can be customized based on needs of the application or user. In this paper, we explore four high-level research questions:

RQ1. How well can we recognize location-specific on-body gestures using finger- and wrist-mounted sensors?
RQ2. Which locations and gestures can be recognized most reliably using this sensing approach?
RQ3. What tradeoffs must be considered when designing and building a realtime interactive on-body input system?
RQ4. How accessible is this approach to users with visual impairments and what are their design preferences?

To address these questions, we evaluated two prototype iterations across two studies. In Study I, we demonstrate feasibility through a controlled data collection study with 24 sighted participants who performed touch-based gestures using the first iteration of our prototype (*TouchCam Offline*). In offline experiments using classifiers trained per-user, we achieve 98% accuracy in classifying coarse-grained locations (*e.g.*, palm, thigh), 84% in classifying fine-grained locations (*e.g.*, five locations on the palm), and 96% in classifying location-specific gestures. Informed by these results, we built a second prototype with updated hardware and software algorithms to support *realtime* on-body localization and gesture recognition (*TouchCam Realtime*). In Study II, we investigate the usability and potential of the realtime system with 12 blind and visually impaired participants. Our findings validate realtime performance with our target population and highlight tradeoffs in accuracy and user preferences across different on-body inputs.

In summary, the primary contributions of this paper include: (i) two iterations of TouchCam, a novel finger-worn camera system that uses machine learning to detect and recognize on-body location-specific gestures; (ii) a quantitative evaluation of our system's accuracy and robustness across a variety of gestures and body locations; (iii) qualitative observations about the usability and utility of our on-body input approach for users with visual impairments; and (iv) design reflections for on-body gestural interfaces in terms of what locations and gestures can be most reliably recognized across users. While our prototype design is preliminary—we expect that future iterations will be much smaller and self-contained—our explorations build a foundation for robust and flexible on-body interactions that support contextual gestures at multiple body locations via supervised learning. Our primary focus is supporting users with visual impairments; however, our approach could also benefit users with situational impairments (*e.g.*, while walking or conversing) or be applied as an input mechanism for virtual reality systems (*e.g.*, for accurate touch-based input in eyes-free situations). All software code and hardware design files are open sourced and available here: https://github.com/lstearns86/touchcam.

2 RELATED WORK

Our research is informed by work on sensing on-body input, finger-worn input devices, and algorithms for texture classification and camera-based biometrics.

2.1 Sensing On-body Input

As noted in the Introduction, on-body input has several advantages over handheld or wearable touchscreen input (*e.g.*, smartphones or smartwatches) offering a larger input surface and more precise touch input even without visual cues [18,45]. However, how to sense this input and what form it should take are still open questions. Researchers have investigated a wide variety of wearable sensing approaches, including cameras [4,10,19,21,50,54,57], IR [27,40–42], ultrasonic rangefinders [30,32], bio-acoustics [20,28], magnetic fields [6], electromyography (EMG) [35], electromagnetic phase shift [62], and capacitance sensors [30,35,48,58]. These approaches support a similarly wide variety of inputs, including discrete touches at different body locations [28,35,48], continuous touch localization on the hand or wrist similar to touchscreen input [4,21,50,62], and input based on 3D finger or arm positions [4,48,50]. We summarize a subset of this prior work alongside our own in Table 1, which helps highlight the diversity of sensing approaches and on-body interaction support. While these past approaches are promising, their sensor types and placements limit the types of interactions that they can support.

First, the interaction space is often constrained to a small surface (*e.g.*, wrist or arm) or to a narrow window in front of the user. Approaches using cameras mounted on the upper body (*e.g.*, [10,19,21,54]) restrict interactions to a pre-defined region within the camera's field of view. *OmniTouch* [21], for example, can only detect gestures on the hands or arms in a relatively small space in front of the user. Similarly, approaches using sensors mounted on one wrist or hand to detect gestures performed by the other hand (*e.g.*, [4,20,27,28,30,41,50,57,58,62]) limit on-body interactions to a relatively small area around the sensors. Some approaches such as *Touché* [48] or *iSkin* [58] are more flexible but still require instrumentation at the target interaction location, which limits scalability. In contrast, our approach places sensors on the *gesturing* finger, supporting input at a variety of body locations within the user's reach without requiring additional instrumentation. Further, while not evaluated in this paper, our system could be readily extended to interact with surfaces beyond the body.

Second, prior work attempts to either identify touched body locations or detect motion gestures but not both. For example, *Touché* [48] and *Botential* [35] can localize touch input at various locations on the body using EMG or capacitance sensors. However, these systems cannot recognize relative surface gestures such as directional swipes. In contrast, systems such as *PalmGesture* [57], *SkinTrack* [41], or *WatchSense* [50] can estimate precise 2D touch coordinates, enabling complex gesture interactions like shapes. However, these methods require sensors affixed on or near the interaction surface to achieve such precision, and they therefore cannot easily be extended to recognize multiple locations. Our approach uses a small camera to identify touched locations, augmented by inertial and IR sensors for robust gesture recognition; together, these sensors enable location-specific gestures.

2.2 Finger-worn Input Devices

TouchCam is worn on the user's gesturing finger and is informed by the growing space of finger-mounted wearables (see [49] for a review). These devices come in many different form-factors (*e.g.*, rings [2,38,43,51], nails [6], below the finger pad [61], or between the fingers [4]), and are used for many different purposes (*e.g.*, reading [51], object/face identification [4,38], or gestural input [2,4,26,43]). For example, *FingerPad* [6] uses a paired magnet and magnetometer on the back of the user's thumb and finger to enable subtle 2D gestural input, while *Light Ring* [26] uses a small gyroscope and IR sensor embedded in a ring to enable input on a variety of surfaces. We focus on touch-based input rather than midair gestures (*e.g.,* [5]) because of the speed and accuracy advantages for non-visual use [45]. We also position our sensors above the gesturing finger to avoid interfering with the user's touch sensitivity and to support flexible input at many different locations.

Most closely related to our work are systems that include finger-worn cameras. *EyeRing* [38], for example, uses a camera ring to identify physical objects in the surrounding environment while *HandSight* [13,51,52], our own prior work, uses a tiny camera atop the finger to assist users with visual impairments in reading and exploring printed text. *CyclopsRing* [4] uses a fisheye lens camera clipped between the fingers to detect one-handed gestures or touch gestures from the user's other hand; however, as discussed above, it cannot easily extend to other locations to support location-specific gestures. *Magic Finger* [61], which partially inspired our approach, uses a tiny camera and an optical mouse sensor worn below the pad of the finger to support touch input on almost any surface, as well as texture-based identification of that surface. However, only two body locations were evaluated and location-specific gestures were not investigated.

Building on the above, we investigate a different set of sensors (optical and inertial), broaden the on-body interaction possibilities (*e.g.,* location-specific gestures), and evaluate a larger number of on-body locations (15

System Name	Sensor type	Sensor placement	On-body Interaction Space	Interaction type
OmniTouch [21]	Camera (Depth)	On the shoulder	On or above the hands or arms (limited by camera FoV)	Continuous touch locations
Touché [48]	Capacitive	Flexible (one on wrist, one elsewhere on body)	Flexible (requires the target location to be instrumented)	Discrete touch locations, body or hand pose
CyclopsRing [4]	Camera (RGB, Fisheye Lens)	Between fingers of passive hand (for on-body input)	On or above the instrumented hand	Continuous touch locations, touch gestures, hand pose
Botential [35]	EMG, capacitive	On the wrist (or arm, leg)	Flexible, different body parts	Discrete touch locations
ViBand [28]	Bio-acoustic	On the wrist	On the instrumented hand or arm	Discrete touch locations, non-directional gestures
SkinTrack [62]	Electromagnetic phase shift	On the wrist, ring on opposite hand	On the skin surface around the instrumented wrist	Continuous touch locations, touch gestures
WatchSense [50]	Camera (Depth)	On the wrist, facing toward fingers	On or above the instrumented hand (limited by camera FoV)	Continuous touch locations, touch and midair gestures
TouchCam (Our Work)	Camera (Grayscale), IMU, IR	On top/side of the gesturing finger and wrist	Flexible (does not require additional instrumentation)	Discrete touch locations, touch gestures

Table 1. Overview of several recent on-body input approaches alongside our own work.

vs. 2 in Magic Finger, for example). We also evaluate our approach with visually impaired participants to explore their specific needs and behaviors, which may differ from sighted users.

2.3 Texture Classification and Camera-Based Biometrics

Work in biometrics has demonstrated that the skin of the hand—specifically, the palm and fingers—contains many distinctive visual features that can be used to identify individuals [7,36]. While skin features are commonly used for identification, they can also support on-body localization—as demonstrated by our initial work [53]. In [53], we extended biometrics algorithms, including those for partial fingerprint and palmprint recognition [12,24] as well as techniques that use webcam or mobile phone camera images for identification and verification [7,9,37,60]. We classified images of small patches of skin on the hand and wrist (patch size ~1-2 cm^2) captured with a handheld camera and achieved high accuracy (>96% across 17 body locations). Our algorithms relied primarily on a texture classification approach inspired by Magic Finger [61] with a local binary pattern (LBP) texture representation and a support vector machine (SVM) classifier. However, as noted in the introduction, our prior work was not wearable, did not function in real-time, only supported localization and not gesture recognition, the data was collected under carefully controlled conditions, and no user studies were conducted. This paper addresses each of those limitations.

3 TOUCHCAM OFFLINE: INITIAL WEARABLE PROTOTYPE

We describe our first prototype, TouchCam Offline, which we evaluate offline using data collected from a controlled study. Study I focuses on addressing RQ1 and RQ2: *how accurately can we recognize location-specific on-body gestures* and *which locations and gestures can be recognized most reliably?* Our results inform the development of a realtime prototype, which is evaluated in Study II (Section 5).

3.1 Prototype Hardware

The TouchCam Offline hardware consists of: (i) a finger-worn multi-sensor package that includes two infrared (IR) reflectance sensors, an inertial measurement unit (IMU), and a small camera with an adjustable LED for illumination; and (ii) a wrist-worn microcontroller with a second IMU, which simulates a smartwatch and provides additional sensing. The finger-based sensors are mounted on three laser-cut rings and positioned to avoid impeding the user's sense of touch, which is particularly important for users with visual impairments. See Figure 2a.

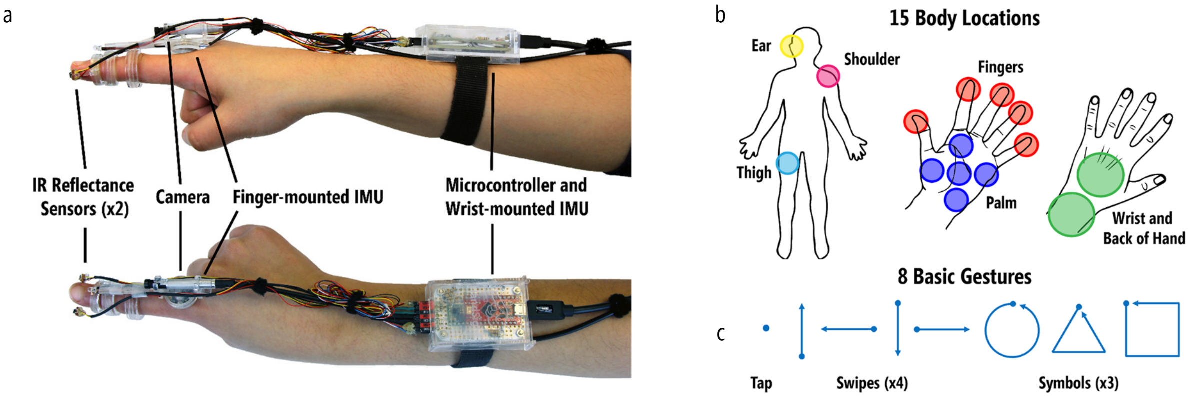


Fig. 2. (a) TouchCam Offline showing the finger and wrist-worn sensors and microcontroller. (b) Fifteen fine-grained body locations (individual circles) within six coarse-grained locations (denoted by color), and (c) eight basic gestures.

Infrared Reflectance Sensors. The two IR sensors[2] (each 2.9 mm × 3.6 mm × 1.7 mm) have a sensing range of ~2–10 mm and are used to detect touch events and to aid in recognizing gestures. Unlike Magic Finger [61], which places optical sensors directly on the pad of the finger, we mount the sensors on the sides of the front-most ring, approximately 5 mm from the fingertip to avoid interfering with tactile sensitivity.

Camera Sensor. A small (6 mm diameter) CMOS camera[3] is mounted atop the user's index finger, providing 640 × 640 px images at up to 90 fps with a 30° diagonal field of view (FOV). We use grayscale images from the camera to classify the touch location, extracting both texture and 2D point features. We also estimate visual motion between video frames to assist in classifying gestures. The camera includes a manually adjustable lens with a focal distance that varies from 15 mm to ∞. Because distances shorter than 30–40 mm provide a very narrow range of focus, we positioned the front of the lens ~50 mm from the user's fingertip. This setup provides an effective FOV of ~27mm across the diagonal when the finger is touching a surface. To prevent lateral rotation and to fix the FOV center near the touch location, the camera is attached to two rings 15 mm apart. A bright LED (3 mm diameter, 6000 mcd, 45° angle) mounted below the camera lens illuminates the touch surface regardless of ambient lighting.

Inertial Measurement Units. Two IMUs[4] are mounted on the user's hand: one below the camera on the index finger and one on the wrist. We include two IMUs to examine the effect of sensor location on classification performance. The IMUs provide motion information at ~190 Hz, and each contains a three-axis accelerometer, gyroscope, and magnetometer. While the camera offers rich contextual information about a scene, its field of view and frame rate limit performance during quick motions. Therefore, the IMUs are our primary sensor for detecting motion and classifying gestures. The orientation of the gesturing finger and/or wrist may also be useful for distinguishing body locations (*e.g.*, ear vs. thigh) although this is posture dependent. The IMUs are calibrated to correct magnetic bias and to establish a stable orientation estimate (described in Section 3.2). The calibration process consists of rotating the unit along each axis for a few seconds and is performed only once per study session—although future explorations may require repeated calibration to ensure long-term stability.

Sensor Placement and Microcontroller. We designed custom laser-cut rings in multiple sizes (13–24 mm inner diameter in 0.5mm increments) with detachable sensors to fit each user. As shown in Figure 2a, the rings are worn on the index finger near the first and second joints. The IR and IMU sensors are controlled via a microcontroller[5] mounted on a Velcro wristband, and the camera and microcontroller are connected to a desktop computer[6] via USB cables. All data is logged, timestamped, and analyzed *post hoc* on a desktop.

3.2 Input Recognition Algorithms

To recognize localized on-body input, we developed a four-stage approach: (i) touch segmentation; (ii) feature extraction; (iii) location classification; (iv) gesture classification. The two classification stages—location and gesture—are trained individually for each user and combine readings from multiple sensors for robustness. While the algorithms described next could be trained on any arbitrary set of locations and gestures, in our study, we evaluated six coarse-grained body locations (*fingers*, *palm*, *back of hand or wrist*, *ear*, *shoulder*, and *thigh)* with 15 fine-grained locations (*thumb/index/middle/ring/pinky finger*, *palm up/down/left/right/center*, *back of hand*, *outer wrist*, *ear*, *shoulder*, and *thigh*) and 8 basic gestures (*tap*, *swipe up/down/left/right*, *circle*, *triangle*, and *square*)—see Figures 2b and 2c. These locations are visually distinctive and can be located easily even without sight while the gestures are simple shapes that can be drawn with a single touch down/up event.

[2] Fairchild Semiconductor QRE113GR

[3] Awaiba NanEye GS Idule Demo Kit

[4] Adafruit Flora LSM9DS0

[5] Sparkfun Arduino Pro Micro (5V/16MHz)

[6] Dell Precision Workstation, dual Intel Xeon CPU @2.1GHz, NVIDIA GeForce GTX 750Ti

Stage I: Touch Segmentation. Our input recognition algorithms receive a sensor stream consisting of video, IMU, and IR data. We segment this input stream by detecting touch-down and touch-up events using the IR sensor readings, which represent distance from the touch surface (lower values are closer). While for real-world use, a segmentation approach would need to identify these touch events within a continuous stream of data, to evaluate this initial prototype we made several assumptions to simplify the process (we eliminated these assumptions for the realtime prototype, described later). Based on experiments with pilot data, we developed a straightforward threshold-based approach using a variable threshold that was set to 90% of the maximum IR value observed across the input stream for each trial. Within a trial, a *touch-down* event is triggered when either of the two IR values crosses below the threshold, while a *touch-up* event is triggered when both cross above the threshold. To be conservative, we assume that each trial contains a single gesture and segment the entire gesture from the first *touch-down* event in the trial to the last *touch-up* event. We crop each input stream to include only the sensor readings and video frames that occurred between the touch-down and touch-up event timestamps.

Stage II: Feature Extraction. In Stage II, we extract static orientation and visual features for localization, and motion features for gesture classification. We describe each in turn below (see Table 4 in the Appendix for more details).

Localization Features. To extract static features for localization, we first determine the video frame that has the maximum focus in the segmented sequence, since it is the most likely to contain recognizable visual features. We define focus as the total number of pixels extracted using a Canny edge detector [3] tuned with a small aperture ($\sigma = 3$) and relatively low thresholds ($T_1 = 100, T_2 = 50$) to detect fine details. While this approach does not account for all image quality problems—motion blur in particular can cause it to fail—it is highly efficient and, in general, detects a much greater number of edges for images that are in focus than for those that are not. We verified this trend empirically using pilot data. We then extract several features for the selected video frame, which include: (i) raw IR sensor readings, (ii) the estimated IMU orientation, (iii) image texture features for coarse-grained classification, and (iv) 2D image keypoints for geometric verification to distinguish between locations with similar textures (*i.e.*, fingertips, palm locations, back of wrist or hand).

The orientation of each IMU is estimated by applying a Madgwick filter [33] on a sequence of raw accelerometer, magnetometer, and gyroscope readings resulting in a 4D orientation vector (*i.e.*, quaternion). The filter is a standard sequential optimization approach to estimating IMU orientations that is updated at each time step. Our initial calibration procedure includes briefly rotating the device in all directions so that the filter can converge to an accurate orientation estimate. The orientation estimate at the selected video frame is used as a 4-dimensional feature vector (*W*, *X*, *Y*, and *Z*) for each IMU and concatenated into an 8-dimensional vector when both IMUs are used.

The image-based features are extracted similarly to our prior work [53]: To represent image texture, we use a variant of local binary patterns (LBP) that is robust to changes in illumination and that achieves rotation invariance while exploiting the complementary nature of local spatial patterns and contrast information [16]. While we explored other common texture-based methods such as Gabor histograms [56] and wavelet principal components [11], we found that they offered negligible improvements over LBP despite their increased computational complexity. We extract uniform LBP patterns and local variance estimates from an image pyramid with eight scales to capture both fine and coarse texture information. Specifically, we use $LBP_{12,2}^{riu2}/VAR_{12,2}$ with 14 uniform pattern bins and 16 variance bins as defined in [16]. These values are accumulated into a histogram with 224 bins for each scale, all concatenated to obtain a 1792-element feature vector. To resolve ambiguities and ensure geometric consistency, we extract custom keypoints at locations with a high Gabor filter response at two or more orientations, which tend to lie at the intersections of ridgelines or creases. This approach was inspired by [23]. We use the Gabor energy in a 16 × 16px neighborhood around the keypoint as a descriptor extracted at multiple orientations to ensure rotation invariance. See [53] for full details.

Motion Features. For gesture classification, we extract motion features from the sensor readings within the segmented timeframe (these are treated independently from the localization features). We use three standard signal preprocessing steps on the raw IMU and IR sensor readings: smoothing, normalization, and resampling. We first smooth the raw values using a Gaussian filter (σ =13, optimized based on pilot data) to reduce the effect of sensor noise and then normalize the smoothed sequence by subtracting its mean and dividing by its standard deviation. To obtain a fixed length sequence for robustness to variations in speed, we resample the sensor readings using linear interpolation at 50 equally spaced discrete time steps. These values, however, are still sensitive to small variations in speed and orientation. Thus, similar to [59], for each IMU and IR sensor we compute summary statistics for windows of 20 samples at 10-step increments (*i.e.,* four windows): mean, minimum, maximum, median, and absolute mean. Finally, for the 50 resampled accelerometer, magnetometer, and gyroscope readings, we compute *x-y*, *x-z*, and *y-z* correlations. The result is 639 features for each IMU and 70 for each IR sensor, which we concatenate into a single feature vector to use when classifying gestures.

We also extract motion features from the video frames between the touch-down and touch-up events. Because we support touch-based gestures only on flat or nearly flat surfaces, it is sufficient to estimate a global 2D motion vector for each frame; we do so using a template-matching approach. First, we down-sample each image from 640 × 640px to 160 × 160px resolution for efficiency and noise robustness. Next, for each frame we extract a 40 × 40px region centered within the previous frame to use as a template, which we then match against the current frame using a sliding window to compute the normalized cross-correlation [29]. The position of the pixel with the highest cross-correlation value identifies the most likely displacement between frames, yielding a 2D motion vector estimate. Because images with higher contrast are more likely to yield reliable motion estimates, we weight each motion vector by an estimate of the frame's contrast (the image variance). As with the other motion features, we smooth the motion estimates by applying a moving average (window size = 10). We then re-sample 50 points from this sequence of motion vectors and compute summary statistics as with the IMU and IR sensor readings to obtain a fixed-length vector of 140 features for use in gesture classification.

Stage III: Localization. Once we have extracted localization and motion features, we begin independently classifying on-body locations (Stage III) and gestures (Stage IV). For localization, we rely primarily on static visual features from the camera with IMU orientations and IR reflectance values to resolve ambiguities.

Our image-based touch localization algorithms function identically to our prior work [53]. We use a two-level location classification hierarchy: first classifying the location as one of the six course-grained regions then refining that location estimate where possible to finer-grained regions. In our offline user study (Study I), coarse-grained regions include *fingers*, *palm*, *back of hand or wrist*, *ear*, *shoulder*, and *thigh* while fine-grained regions include specific fingertips, locations on the palm, and on the back of hand versus the wrist (Figure 2b). Some coarse-grained locations are not subdivided at this second level due to a lack of distinctive features—in the case of our study, the *ear*, *shoulder*, and *thigh* are not subdivided. We first classify the texture features into a coarse-grained location using an SVM[7] then perform template matching against only the stored templates from that location to estimate the fine-grained location. Finally, we perform geometric verification using the extracted 2D point features to ensure a correct match.

At both levels of the classification hierarchy, we resolve ambiguities using a sensor fusion approach. We combine predictions based on the static visual features from a video frame with predictions based on the IMU orientation and IR reflectance features with the same timestamp as that frame. Since the scales, lengths, and types of these feature vectors are all very different, rather than concatenating the features for use with a single classifier we instead train a separate SVM with a Gaussian kernel on the non-visual features. To robustly combine the predictions from the two disparate localization classifiers (one for the camera features and one for the IR and IMU features), we first tune the SVMs to output normalized probability predictions for each class

[7] Aforge.NET: http://www.aforgenet.com (used for all SVM and neural network classifiers)

using Platt scaling, as is standard [47]. We concatenate these predictions into a single feature vector, which we then use to train a third sensor fusion classifier that automatically learns how to prioritize sensors based on prediction confidence and location class. Inspired by [8], we use a feedforward neural network for this sensor fusion classifier. Our network has one fully connected hidden layer for flexibility of functional representation, and a softmax output layer for multiclass output; it is trained using resilient backpropagation [34]. The final output of our classification process is a combined location prediction from among the six coarse-grained and fifteen fine-grained classes with approximate likelihoods for each class (sorted from most to least likely).

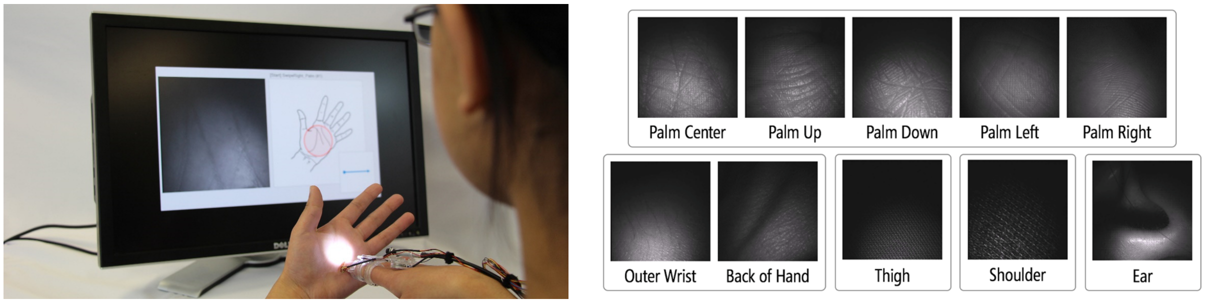
Stage IV: Gesture Classification. Gesture classification is performed independently of localization using an additional SVM. As with texture, SVM classifiers are commonly used for classifying gesture features because they are robust and efficient for problems with high dimensionality. We use a linear kernel with feature weights that were optimized for performance across all participants. For the evaluation presented in Section 3.4, we trained an SVM to classify the following gestures as shown in Figure 2c: *tap*, *swipe up*, *swipe down*, *swipe left*, *swipe right*, *circle*, *triangle*, and *square*.

3.3 Study I: Data Collection and Dataset For Offline Experiments

To evaluate our initial prototype and algorithms, we performed offline experiments using data collected from twenty-four participants. Each participant performed a series of location-specific on-body input tasks with our hardware prototype. We were specifically interested in investigating our first two research questions enumerated in the Introduction: (i) How accurately can we recognize location-specific on-body gestures with a finger-worn camera and auxiliary sensors (IMU, IR)? (ii) Which body locations and gestures can be recognized most reliably using our approach?

Participants. Twenty-four right-handed participants (16 female) were recruited via campus e-mail lists and word of mouth. Their average age was 28.9 (*SD*=7.95, *range*=19–51). All participants had normal vision as the goal of this study was to assess our algorithms and not issues related to usability or accessibility. Participants were compensated $25 for their time.

Data Collection Apparatus. During data collection, participants wore the TouchCam Offline prototype. As described in Section 3.1, we selected ring sizes to fit the participant's finger and adjusted positioning to ensure a consistent sensor range. A custom application written in C# displayed visual task prompts and a live feed from the finger-worn camera to assist with framing the target locations (Figure 3a). All IMU and IR sensor readings and camera video frames were logged with timestamps along with ground-truth touch location and gesture labels for each trial.



(a) Participant following on-screen data collection protocol

(b) Example skin images from Study I

Fig. 3. (a) Data collection setup showing our prototype, location and gesture instructions, and camera video feed. (b) Example skin-surface images recorded by our finger-mounted camera (fingerprint images omitted to protect our participants' privacy).

Procedure. The procedure lasted up to 90 minutes. After a brief demographic questionnaire and setup period (*i.e.*, selecting rings, putting on the prototype), participants completed the following tasks, in order:

1. *Location-specific touches.* Participants touched and held their finger in place at 15 locations (Figure 2b) with each location prompted visually on a monitor (Figure 3a). After confirming the location and image quality, the experimenter logged the current location (*e.g.*, timestamp, location label) and triggered the start of the next trial. Participants completed 10 blocks of trials, where each block consisted of a different random permutation of the 15 locations (150 trials in total). In total, this dataset includes 3600 *location-specific touches* across all participants. Example images are shown in Figure 3b.
2. *Location-specific gestures.* Participants performed the eight basic gestures: *tap*, *swipe up*, *swipe down*, *swipe left*, *swipe right*, *circle*, *triangle*, and *square* (Figure 2c) at three body locations: the *palm*, *wrist*, and *thigh*. These locations were selected from the 15 locations in the first task because they are easy to access, unobtrusive, and have a relatively large input area thus allowing for more complex gestures. As with the first task, participants completed 10 blocks of trials, where each block consisted of a different random permutation of the 24 gesture and location combinations (240 trials in total). This dataset includes 5,760 *location-specific gestures* across all participants.

3.4 Study I: Offline Experiments and Results

To investigate the accuracy of our location and gesture classification algorithms, we performed a series of offline experiments using the gathered data. Below, we evaluate coarse-grained localization, fine-grained localization, and location-specific gesture classification as well as the effect of each sensor on performance (*e.g.,* finger-worn *vs.* wrist-worn IMUs). We compare sensor combinations using paired t-tests and Holm-Bonferroni adjustments to protect against Type I error [22].

Training and Cross Validation. All of our experiments use leave-one-out cross validation and train and test on a single user's data. Specifically, each experiment uses all available data from a single participant for training the location and gesture classification SVMs with a single sample set aside for testing. The localization and gesture classifiers are trained independently. The experiment is repeated for each sample and averaged across all possible combinations.

Touch Localization. To examine the accuracy of our on-body localization algorithms, we used the location-specific touch dataset. Since our localization approach is hierarchical, we analyze performance at both the coarse-grained level (6 classes) and the fine-grained level (15 classes).

We first report primary localization results using all available sensor readings (*i.e.,* sensor fusion results). At the coarse-grained level, we achieve 98.0% (*SD*=2.3%) average accuracy. This is reduced to 88.7% (*SD*=7.0%) at the fine-grained level. Table 2 shows the accuracy breakdown by class. The worst performing coarse-grained classes were the wrist/hand and ear, both at 93.8%, possibly due to their highly variable appearance and fewer distinctive visual features. In contrast, the fingers and palm perform best at 99.6% and 99.1% respectively although the individual fine-grained classification accuracies were lower. These results suggest that care must be taken in selecting body locations that are both visually distinctive and easy for participants to return to repeatedly. A qualitative analysis of our dataset revealed issues that account for some of the error: approximately 5% of the images gathered had focus, contrast, or illumination issues that interfered with

	Palm	Fingers	Wrist/Hand	Ear	Shoulder	Thigh
Palm	99.1%	0.5%	0.4%			0.1%
Fingers	0.3%	99.6%				0.1%
Wrist/Hand	5.0%	0.2%	93.8%	0.2%	0.6%	0.2%
Ear	4.2%	0.4%	1.2%	93.8%	0.4%	
Shoulder	0.8%		0.4%		98.8%	
Thigh	2.3%		0.4%			97.3%

(a) Coarse-grained Accuracy

	Up	Down	Left	Right	Center
Palm	84.6%	78.5%	85.0%	83.1%	91.5%

	Thumb	Index	Middle	Ring	Pinky
Fingers	93.1%	85.4%	81.5%	88.1%	91.9%

Outer Wrist	Back of Hand	Ear	Shoulder	Thigh
87.3%	88.8%	93.8%	98.8%	97.3%

(b) Fine-grained Accuracy

Table 2. Classification percentages averaged across 10 trials and 24 participants. (a) Accuracy for the six coarse-grained classes. Each cell indicates the percentage of images assigned to a predicted class (column) for each actual class (row); empty cells indicate 0%. (b) Accuracy for the 15 fine-grained classes, grouped by corresponding coarse-grained class.

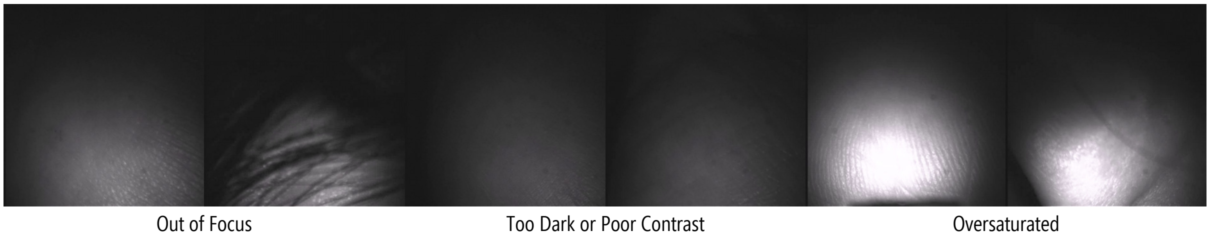


Fig. 4. Approximately 5% of the images we collected had poor focus, contrast, or illumination, preventing robust feature extraction. We adjusted the camera and LED to mitigate these issues for TouchCam Realtime.

extracting recognizable image features; see Figure 4. We took steps to mitigate these problems for the next iteration of TouchCam.

To investigate the effect of each sensor on localization performance, we repeated the classification experiment with the sensors individually and in combination. As expected, the camera is by far the most accurate single sensor for classifying location, with a coarse-grained accuracy of 97.5% (*SD*=2.6%) followed by the finger-based IMU at 75.6% (*SD*=11.6%). Notably, the camera is significantly better even compared to the 87.5% (*SD*=7.0%) accuracy of combining all other sensors (p<0.001, t_{23}=7.12, d=1.92). No significant differences were found between the camera alone or combined with other sensors, which suggests that the camera alone is sufficient for course-grained classification. At the fine-grained level, the camera is again the most accurate sensor (84.0%) even compared to all other sensors in combination (52.9% accuracy; *SD*=12.0; p<0.001, t_{23}=16.74, d=2.99). But, unlike at the coarse-grained level, adding any of the other three sensors to the camera further increases accuracy, with the highest accuracy (88.7%) resulting from the combination of all available sensors.

Location-Specific Gesture Classification. To explore the possibility of supporting location-specific gestures, we conducted a classification experiment with the data from the location-specific gesture task (24 classes: 3 locations × 8 gestures). First, we classified the location using the image features from the camera (extracted from the video frame with maximal focus as described above). Since the location features from the IR and IMU sensors did not make a significant difference at the coarse-grained level, we omitted them here. Location accuracy for these three locations was 99.1% (*SD*=1.0%). Next, we classified the gesture using the motion features from all of the sensors (IMU, IR, and camera) achieving an accuracy of 96.6% (*SD*=2.6%). Finally, we combined the classification predictions and calculated the overall location-specific gesture classification accuracy across all 24 classes, which was 95.7% (*SD*=3.2%).

As a secondary analysis, we again examined classification accuracy as a function of each sensor (Figure 5) but this time for the 24 location-specific gestures. In general, adding more sensors significantly improves classification accuracy, although as a practical matter the differences between the pair of IMUs and other more complex combinations are fairly small (see the Appendix for statistical comparisons).

Efficiency. For our initial prototype and algorithm development, our primary aim was to investigate the feasibility and accuracy of our approach rather than develop a realtime system. As such, our TouchCam Offline algorithms are slow. On our desktop computer (the Dell Precision Workstation described in Section 3.1), the

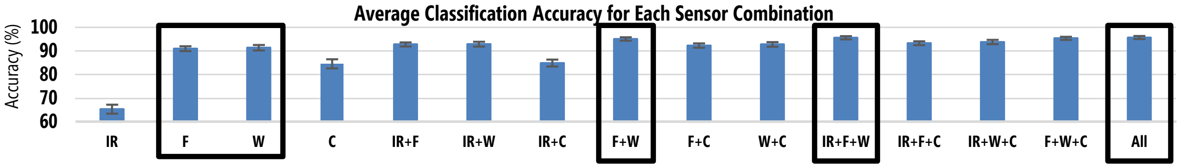


Fig. 5. Mean classification accuracy using different sensor combinations to classify location-specific gestures. Boxes indicate the best sensor combinations as additional sensors are added, with each box significantly outperforming the last (from left to right). There was no significant difference between the finger- and wrist-mounted IMUs.

image feature extraction and localization stages required, on average, two seconds *per frame* to process and classify an image. The most computationally demanding stage was the geometric verification process, which required approximately 243,000 feature comparisons on average. The other stages' computation times are comparatively negligible.

3.5 Summary of Study I Findings

Our results address our first two research questions demonstrating the feasibility of recognizing location-specific gestures using finger- and wrist-worn sensors. While our experiments show advantages with sensor fusion when classifying both location and gesture, the practical differences are relatively small suggesting that we can simplify our algorithms by using each sensor type for the task for which it is best suited (*i.e.*, IR sensors for touch detection, camera for localization, and IMUs for gesture recognition). Individual accuracies per location suggest limits to the localization granularity of our algorithms, which performed well (≥98%) for coarse-grained locations but were less accurate (89%) for fine-grained locations. These results could likely be improved with better camera hardware (*e.g.,* higher resolution, autofocus) and with more complex finger/palm print recognition algorithms. However, the high accuracy during our location-specific gesture experiment (96%) suggests that such steps may not be necessary for us to begin investigating these interactions with visually impaired users. We built upon these findings to implement the next iteration of TouchCam, described below.

4 TOUCHCAM REALTIME: IMPROVED INTERACTIVE PROTOTYPE

Based on our Study I findings, we designed *TouchCam Realtime*, a realtime version of our offline system with updated hardware and algorithms. We first describe key changes to improve robustness and enable realtime interactions (addressing RQ3) before validating the new classification algorithms using the Study I data.

4.1 Realtime Prototype Hardware

TouchCam Realtime's hardware (Figure 6) embeds all finger-mounted components in a single unit, which is attached to the user's finger using a pair of Velcro strips to allow greater freedom of motion than the rigid rings from the previous version. This updated design is more stable and durable. The camera and IR sensors are repositioned to capture more consistent images and improve the reliability of touch detection, respectively. Although Study I found an accuracy advantage when using two IMUs (~4%), we decided to remove the wrist-mounted IMU to simplify our hardware and algorithms. We compensated for the potential drop in accuracy by doubling the remaining (finger-mounted) IMU's sampling rate. This change reduced the number of features used to classify gestures and the number of examples needed for training.

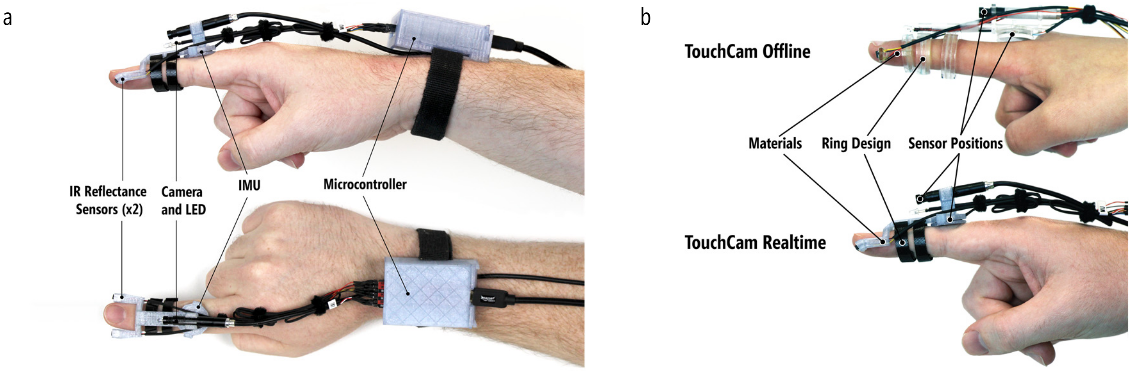


Fig. 6. (a) TouchCam Realtime prototype showing the finger and wrist-worn sensors and wrist-worn microcontroller. (b) Comparison between TouchCam Offline and Realtime hardware.

4.2 Realtime Input Recognition Algorithms

We made several changes to our input recognition algorithms to support realtime operation. First, we optimized our localization algorithms to run in realtime on a GPU and removed the computationally costly geometric verification step. Second, we updated the touch detection stage to support continuous use. Finally, we improved the gesture recognition stage, making it more robust to changes in orientation and pose.

Localization Algorithm Changes. As noted previously, our offline localization algorithms required up to two seconds per frame, primarily limited by geometric keypoint matching between image templates. Simply removing keypoint matching increased our frame rate from 0.5fps to ~18.5fps, but with a ~9% reduction in Study I's fine-grained localization accuracy. To address this loss, we made three updates to our localization algorithms. First, we used an alternate LBP algorithm that better preserved spatial features [64], which increased the number of texture features per image from 1792 to 15,552. Second, we averaged class probability predictions across 20 video frames, a number selected after pilot tests to balance accuracy and latency. And third, we reduced the number of fine-grained locations, omitting the five fingertip locations evaluated in Study I. This decision was not solely due to algorithmic performance—the fingertips proved difficult for participants to capture reliably even with visual feedback due to the sensors' positioning and small field of view. Also, while the fingertips are convenient locations for static touch-based input, they are too small to easily support gestural input. Finally, we implemented parallel GPU versions of our algorithms, which further improved the localization speed to 35.7 fps.

Touch Detection Algorithm Changes. To improve robustness and support continuous use, we made minor changes to the touch detection algorithms. We applied a moving average filter to the IR values to reduce sensor noise (*window size* = 50ms), and triggered *touch-down* and *touch-up* events when the sensors crossed a fixed threshold that was the same across all users rather than derived per gesture as with the offline system. This threshold was fixed at 90% of the maximum possible value the sensor could register, which we determined empirically to be robust to changes in ambient lighting and to work well for skin and clothing surfaces. To ensure that we captured the full gesture (and to support the double-tap gesture), we placed a delay of 100ms on the *touch-up* event and canceled it if the user touched down again within that period.

Gesture Recognition Algorithm Changes. Lastly, we made improvements to the gesture recognition algorithms. To compensate for variations in orientation and pose when performing gestures, we first rotated the IMU sensor readings relative to the estimated orientation at the start of the gesture (the *touch-down* event). We discarded the magnetometer readings after this step since they were still overly sensitive to orientation and location. These changes allowed us to build a pre-trained cross-user gesture classifier with 1,720 samples in place of the individual classifiers used in Study I.

4.3 Validation of Realtime Algorithms

To test our updated algorithms and establish a performance benchmark for our realtime system, we conducted classification experiments on the data gathered during Study I. The average 10-fold cross-validation accuracy on the location-specific touches dataset was 97.5% (*SD*=2.4%) at the coarse-grained level (6 classes) and 84.5%

	Palm	Fingers	Wrist/Hand	Ear	Shoulder	Thigh
Palm	98.5%	0.8%	0.5%	0.1%		0.1%
Fingers	0.3%	99.7%				
Wrist/Hand	4.0%	0.4%	95.4%			0.2%
Ear	5.4%	2.1%		92.1%	0.4%	
Shoulder	1.7%	0.4%	2.1%		95.4%	0.4%
Thigh	1.7%		2.1%	0.4%		95.8%

(a) Coarse-grained Accuracy

	Up	Down	Left	Right	Center
Palm	83.8%	82.5%	80.8%	85.4%	89.6%

	Thumb	Index	Middle	Ring	Pinky
Fingers	92.1%	71.3%	71.3%	73.8%	79.6%

Outer Wrist	Back of Hand	Ear	Shoulder	Thigh
87.5%	85.8%	92.1%	95.4%	95.8%

(b) Fine-grained Accuracy

Table 3. TouchCam Realtime performance on Study I dataset. (a) Coarse-grained classification percentages, averaged across 10 trials and 24 participants. Each cell indicates the percentage of images assigned to a predicted class (column) for each actual class (row). (b) Fine-grained classification percentages, averaged across the corresponding coarse-grained classes.

(*SD*=8.2%) at the fine-grained level (15 classes), which is nearly identical to the TouchCam Offline system—see Table 3. The five finger locations were most impacted by the removal of the geometric verification step. Localization accuracy on the location-specific gestures dataset remains similarly high at 98.6%. As mentioned above, efficiency increased considerably: from 0.5fps to 35.7fps (a ~70x speedup).

5 STUDY II: REALTIME EVALUATION WITH VISUALLY IMPAIRED PARTICIPANTS

To assess the performance and accessibility of TouchCam Realtime under more realistic conditions and with our target population (RQ4), we conducted a second study. We recruited 12 blind and visually impaired participants who performed common interactions with TouchCam such as checking the time or reading text messages. We focus primarily on issues impacting the accuracy and usability of our system (see [46] for more about the interaction designs and participant feedback).

5.1 Study II: Method

Participants completed an adaptive calibration procedure for training and then used TouchCam Realtime to perform tasks using three on-body interaction techniques.

Participants. Twelve participants (7 female, 5 male) were recruited through email lists, local organizations for people with visual impairments, and word of mouth. Nine participants were blind and three had low vision. The average age was 46.2 years old (*SD*=12.0, *range*=29–65). All participants were smartphone users (11 iPhone, 1 Android) and all reported using a screenreader either “all” or “most” of the time. Participants were compensated $60 for time and travel.

Apparatus. Throughout the study, participants wore the TouchCam Realtime prototype on their dominant hand. We assisted participants with putting on the ring and wristband and adjusted positioning to ensure consistent sensor readings. A custom C# application controlled a semi-automated training process, provided audio and synthesized speech cues during the tasks, and displayed a camera and sensor view for the researcher to ensure correct positioning. All sensor readings and video frames were logged with timestamps.

Location and Gestures. Based on observations made during Study I, we refined the locations and gestures for Study II. We reduced the coarse-grain set from 6 to 4 locations and the fine-grain set from 15 to 9 locations. Specifically, we replaced the back-of-the-hand location with the inner wrist due to inter-class similarity with the outer wrist, removed the shoulder location for ergonomic reasons, and removed the five finger locations because without 2D keypoint matching and geometric verification, classification accuracy for this region was considerably lower. The updated set of locations included: the palm (up, down, left, right, and center), the wrist (inner and outer), the thigh, and the ear (Figure 7).

While Study I showed that TouchCam can support a variety of touch-based gestures, for Study II we specifically modeled our interactions after Apple’s VoiceOver[8] and Google’s TalkBack[9]—two popular gesture-based mobile screenreaders for non-visual use. In total, we support 6 gestures, including: *swipe left* or *swipe right* to move between menu items and *double-tap* to select an item. We also included a *single-tap* gesture to repeat a voice prompt, a *swipe-down* gesture to go to the previous menu, and a *tap-and-hold* gesture to select

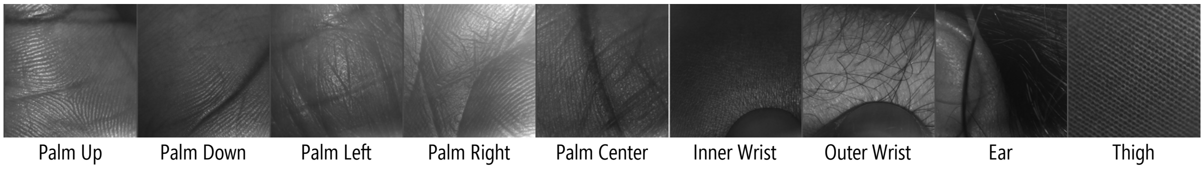


Fig. 7. Sample image data collected with TouchCam Realtime. All images selected from different participants.

[8] http://www.apple.com/accessibility/ios/voiceover/

[9] https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback

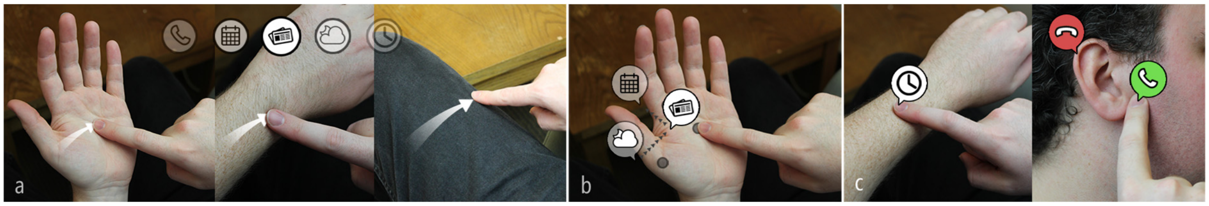
location-specific items. The tap-and-hold gesture was recognized by an 800ms timeout after the *touch-down* event while the other gestures were recognized using a pre-trained SVM classifier (as described in Section 3.2). These gestures can be performed at any body location.

Training Procedure. To limit the amount of time needed to train our system, we implemented an adaptive training procedure inspired by boosting [14]. After capturing a single image of each of the nine locations for initialization, participants then moved their finger around each location in a fixed order as the system continuously classified the video frames. Whenever a video frame was misclassified, that frame and the current location label were saved and the classifiers were retrained. This semi-automated training continued until convergence (*i.e.*, until the researcher determined that the automated system was performing well). After training all locations, at least one additional round was necessary to ensure that new image samples did not negatively affect performance. We found that the initial training images plus two rounds of semi-automated training were sufficient for most users, which took roughly 15-20 minutes and resulted in an average of 13 training examples per location (*SD*=4.5; *range*=5-24).

Procedure. The study procedure lasted up to two hours, and consisted of: (i) an interview about mobile and wearable device usage including thoughts about on-body interaction (~20 minutes); (ii) system calibration and training (~30 minutes); (iii) using TouchCam with three interaction techniques (~10 minutes each); and (iv) a post-study questionnaire (~15 minutes). For (iii), the VoiceOver-like interaction techniques were presented in a fully counterbalanced order. Each interaction technique supported the same set of applications and menu items accessed through a two-level hierarchical menu. The top-level menu had five applications (*Clock*, *Daily Summary*, *Notifications*, *Health and Activities*, and *Voice Input*), which were selected by double tapping. Once selected, each application had 3-4 submenu items except for *Voice Input*, which had no submenu. The three interaction techniques are described below (see also: Figure 8 and the supplementary video):

1. *Location-independent gestures (LI).* Users swiped left or right anywhere to select an application.
2. *Location-specific palm gestures (LS_{palm}).* Top-level applications were mapped to five different locations on the palm. Users pointed directly to a location to select that application or searched for an item by sliding their finger between locations (similar to VoiceOver).
3. *Location-specific body gestures (LS_{body}).* Functioned similarly to LS_{palm} but mapped the applications to five different locations on the body rather than just the palm. We attempted to use intuitive mappings. For example, tapping the outer wrist for Clock and the ear for Voice Input. The other mappings were: the palm for Notifications, the inner wrist for Daily Summary, and the thigh for Health and Activities.

After activating an application, navigation of the submenus was identical across all three interaction techniques, using swipes left and right to select an item and a double-tap to activate it. For each of these interaction techniques, participants were instructed to complete the same set of 10 tasks in a random order. After an automated voice prompt said "begin," a task consisted of selecting an application, opening its submenu, and then selecting and activating a specific menu item (*e.g.*, "open the *Alarm* item under the *Clock* menu"). After



(a) Location-independent gestures (*LI*) **(b) Location-specific palm gestures (*LS_{palm}*)** **(c)Location-specific body gestures (*LS_{body}*)**

Fig. 8. Three on-body interaction techniques: (a) for *LI*, users swipe left/right anywhere on the body to select an application. For (b) and (c), users select an application by double tapping on a specific location on their palm (LS_{palm}) or body (LS_{body}).

the correct menu item had been activated by double-tapping, an automated voice prompt said "task complete" and participants proceeded to the next task.

The session concluded with open-ended questions about the participant's experience using TouchCam Realtime and the three interaction techniques.

Data and Analysis. Throughout the study, we logged all sensor readings, the location and gesture classifications, and event occurrences (*e.g.,* task start/end, menu navigation). We analyze performance in terms of classification accuracy, as well as qualitative metrics of robustness and usability for the three on-body interaction techniques that we tested. We also describe qualitative reactions and subjective preferences based on the interviews and questionnaires.

5.2 Study II: Experiments and Results

To evaluate TouchCam's realtime performance and usability with our blind and low-vision users, we observed participants' behavior during the study and analyzed subjective feedback about our system. We also conducted offline experiments as with Study I, focusing on the sensor data gathered during the training phase of the study (rather than later data, which was unlabeled). Below, we summarize the details of our experiments and findings.

General Observations and Reactions. All twelve participants successfully used TouchCam Realtime to complete tasks with each of the three interaction techniques. In the pre-study interview, most participants (*N*=9) reacted positively toward the idea of on-body interaction citing quick and easy access (N=7), the ability to map specific tasks to different body locations (*N*=6), reducing the number of devices to be carried (*N*=6), and not needing to hold a phone in hand, thus avoiding the risk of theft or damage and potentially freeing that hand for other tasks (*N*=4).

Participants reacted similarly after using the TouchCam prototype. Preferences were split between the three interaction techniques. Participants appreciated the low learning curve and flexible input location of the *LI* interface, which supported simple swipe and tap gestures anywhere on the body, while the location-specific LS_{palm} and LS_{body} interfaces offered quicker and more direct selections once the location mappings were learned. Some participants preferred the proximity of locations for LS_{palm} because it enabled easy exploration and minimal movement, while others liked the more intuitive location mappings of LS_{body}. Key concerns included TouchCam's large physical size, the occasional difficulty with the LS_{palm} interface due to its lower fine-grained accuracy, and the social acceptability of using LS_{body} in public (*e.g.,* touching an ear may draw unwanted attention to the device). See [46] for a more thorough examination of qualitative reactions to our system.

Localization Accuracy. To assess TouchCam's localization accuracy and robustness for visually impaired users, we analyzed the data gathered during the *training* phase of the study. We first conducted a leave-one-out cross-validation experiment using the recorded training samples for each participant (similar to Study I). This resulted in an average accuracy of 91.2% (*SD*=3.5%) at the coarse-grained level and 76.3% (*SD*=76.3%) at the fine-grained level, which is a drop in performance compared to Section 4.3. This decrease, however, is reflective of our adaptive training procedure: since new samples are added only when misclassified using the current SVM, we would naturally expect lower performance when removing even a single sample for cross-validation.

Thus, we conducted an additional experiment using the full training set and classified other video frames from the training session (*i.e.*, those recorded in between the stored training samples). Here, the accuracy increases to 94.2% (*SD*=5.0%) and 81.3% (*SD*=6.6%) respectively. These latter numbers better reflect actual usage performance since we could not reliably measure ground truth during the actual user study (*i.e.,* when participants were using TouchCam with the three interaction techniques). We note that though performance should be improved in future work (see Discussion), these results were sufficient for using and evaluating TouchCam with our participants.

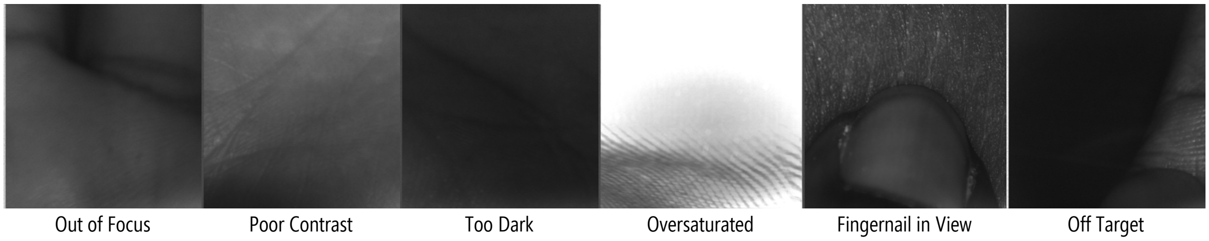


Fig. 9. Some images captured during Study II were of poor quality due to the highlighted reasons. Despite these issues performance remained adequate for participants to complete our specified tasks.

Robustness. To investigate this drop in performance in more detail, we performed a manual inspection of the 1,380 training images across the 12 participants using a custom image reviewing tool. While the severity of the problems varied widely, 22.2% of the images had some issue that could interfere with reliable classification (Figure 9), including: poor focus (13.6%), insufficient illumination (5.4%), poor contrast (4.3%), or oversaturation (0.8%). In addition, 3.2% of the images did not capture the target location due to the offset between the participant's touch location and the center of the camera's field of view, and in 0.6% of the images the participant's finger filled a large portion of the field of view, reducing the number of pixels available for identifying the target location. We further discuss robustness in the Discussion.

5.3 Summary of Study II Findings

Our findings validate TouchCam Realtime's performance with our target population and demonstrate three possible on-body interaction techniques that our approach can support. Participants successfully performed several simple input tasks with our system, and their comments highlight positive reactions to on-body input as well as tradeoffs between the three interaction techniques. These tradeoffs reflect both TouchCam's performance (*e.g.*, LS_{palm} was least accurate due to its reliance on fine-grained localization) and broader design implications (*e.g.*, user preferences for flexibility of input location, learning curve, and social acceptability). Our findings also highlight obstacles to robust on-body input recognition, especially for visually impaired users who cannot rely on visual cues.

6 DISCUSSION

While prior work has explored preliminary issues related to the design of on-body interfaces for visually impaired users [18,44], TouchCam is the first wearable on-body input system supporting real-time interaction designed for and evaluated with this population. Moreover, our work contributes the first real-time system for localizing skin images, and the first to explore location-specific touch-based gestures at a wide set of body locations. Below we discuss TouchCam's performance and usability and provide recommendations for future on-body input systems to support users with visual impairments.

6.1 Robust On-Body Input Detection Using Sensors On the Gesturing Finger and Wrist

Because TouchCam's sensors move with the gesturing finger, they can support touch input at a variety of body (and non-body) locations without requiring additional instrumentation. This feature allows greater input flexibility than most other on-body input approaches (*e.g.*, compared to ViBand [28] or Touché [48]) and means that the user is also less likely to encounter issues with camera framing or occlusion—problems that are common for VI users when they use camera-based systems [1]. Although we did not examine non-body interactions in our work, TouchCam should support location-specific gestures at any surface with visually distinctive features.

Our results demonstrate the feasibility of a computer-vision driven finger camera approach for on-body input; however, we also encountered obstacles that limit TouchCam's accuracy and precision. Because of the camera's size and positioning, image quality was variable. A high percentage (22.2%) of the training images gathered during Study II were out of focus, low contrast, or poorly illuminated, and in some images the target location was not visible due to the offset between the participant's touch location and the center of the camera's field of view. These usage issues appeared to have a greater impact on performance than other potential factors such as ambient lighting, skin tone, age, or hand size, although future work should investigate these possibilities in greater detail. Improved camera hardware could help address some problems—for example, autofocus functionality would help ensure sharp focus across changes in camera distance or perspective and a wider-angle lens would provide additional contextual information to aid classification. Audio feedback that notifies users when there is a problem and helps them learn how to use the system, as provided by assistive devices for reading such as *KNFB Reader*[10] or *OrCam*[11], could also be helpful. Finally, future work should explore hybrid sensing approaches that combine a finger-mounted camera with an additional body-worn sensor on the head or chest, which could provide additional contextual information and assist with localization.

6.2 An Expanded On-Body Input Vocabulary

As mentioned above, our work introduces new types of on-body interactions that other systems cannot readily support without additional instrumentation. For example, the fixed sensors used by ViBand's smartwatch platform limit interactions to a relatively small area on the hand and arm [28] while Touché requires modification of the target interaction surface and cannot detect gestural input [48]. In contrast, TouchCam can recognize location-specific gestures at several body locations, potentially allowing for intuitive context-specific input (*e.g.*, tapping the wrist to check the time) and supporting a high degree of flexibility and customization.

Participants identified tradeoffs between our three proof-of-concept interface designs, which should be considered when designing on-body interfaces to strike a balance between speed, accuracy, and learnability. Location-independent gestures (*LI*), which allow navigation using swipe gestures anywhere on the body, are easy to understand and learn, do not require individual calibration, and enable flexible input as needed for different situations (*e.g.*, sitting at home *vs.* walking while holding a cane). Location-specific gestures (LS_{palm} and LS_{body}), where the user can directly select an application or menu item by touching a specific location, are potentially quicker once the location mappings have been learned and can also support intuitive context-specific gestures as mentioned above. The palm-only version (LS_{palm}), with its high touch sensitivity and close proximity between mapped locations, could enable faster and more discrete input. Compared with the other two versions, it also more readily supports "touch and explore" functionality that could help participants learn the location mappings more quickly. However, in our experiments LS_{palm} was less accurate than the other two because of inter-class similarity between palm locations and thus required participants to more carefully position their hand and fingers.

This expanded input vocabulary and flexibility of input locations may come at a cost, at least in the current iteration of TouchCam. While our prior work [53] suggested that we should be able to support precise localization on the palm and fingers using their rich visual features, our findings in this work highlight difficulty with robustly recognizing fine-grained locations. Future work should investigate ways to more reliably recognize fine locations, ideally with greater granularity than tested in our studies (*e.g.*, more than five palm locations), and recognizing touch input at two or more locations simultaneously (*e.g.*, using multiple finger-worn sensors) to support multi-touch gestures. In particular, future work should investigate how to extend our approach to support precise 2D localization (*e.g.*, as with OmniTouch [21] or CyclopsRing [4]). These goals may be possible with the aid of additional sensors (*e.g.*, a body-mounted camera) or with more efficient fingerprint and palmprint recognition algorithms that can support real-time interactions.

[10] http://www.knfbreader.com/

[11] http://www.orcam.com/

6.3 Training and Calibration

While TouchCam's gesture recognition algorithms are robust enough to allow for a shared classifier that works across users, its localization algorithms rely on unique skin and clothing features and must be individually calibrated for each user. This requirement raises two concerns: (i) the time needed to complete the individual training procedure, and (ii) the stability and robustness of the classifiers over time as the system shifts position and the user's body appearance varies (*e.g.*, due to changing moisture levels or clothing). We took steps to address the first concern in Study II by introducing our automated training procedure, which took about 15-20 minutes for a new user compared to 30-45 minutes in Study I. However, this procedure will likely need to be simplified and further streamlined in future versions. One possibility would be to bootstrap the system using a large amount training data across multiple users, which could enable coarse-grained classification without individual training. Fine-grained accuracy could be improved over time by learning as the system is used.

As for the second concern, it is possible (even likely) that shifts in the sensor positions after calibration negatively impacted performance for some participants during Study II. Long-term performance is a challenge for many on-body input systems, since they can be highly sensitive to sensor positioning and biometric changes [63]. To explore how accuracy is affected over time, we conducted a small additional study with data gathered across five identical sessions with a single user (the first author). The time between sessions varied from 15 minutes to 24 hours, with the sessions completed over a three-day period. The prototype was fully removed between each session. Classification accuracy was measured similarly to the other experiments described above, except previous session data was used for training and the current session for testing.

As expected, accuracy drops considerably when training on a single session and testing on another, from the 94.2% coarse-grained and 81.3% fine-grained numbers reported in Study II down to 88.2% and 73.6% on average respectively. However, combining training data across sessions improves accuracy reaching an average of 96.5% and 91.8% at the two levels for four training sessions (Figure 10). A larger longitudinal study will be necessary to determine how well these results extend to other users and to a longer period of time, but these results are promising.

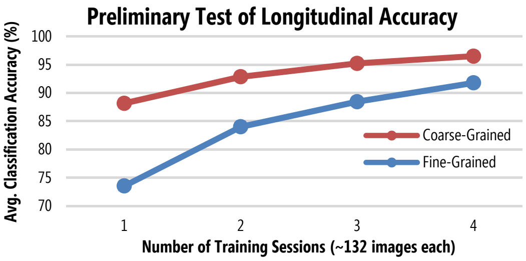


Fig. 10. Classification accuracy across multiple sessions. In general, accuracy increases with more training sessions, suggesting that recalibration may initially be necessary but that accuracy will eventually converge.

6.4 Physical Design

We designed TouchCam to avoid interfering with the user's movements and sense of touch, but the system is still large and requires tethering to a desktop computer for fast processing. With further algorithmic optimizations and increases in mobile processing power, we ultimately envision a smaller, self-contained system that uses a smartwatch for processing and power. Furthermore, our priority with the finger-worn components was to ensure robustness and durability during our experiments, but our design can be streamlined considerably using existing technology. For example, the 6mm diameter camera module[12] that we selected could be replaced with a much smaller 1mm unit from the same manufacturer[13], and the IMU components could be embedded more directly into the ring (while the board we used is 16mm in diameter, the IMU itself is only 4mm square). The IR reflectance sensors positioned near the tip of the user's finger could potentially be replaced with an alternative touch detection method that is less intrusive—for example, an IR depth sensor with a longer range. Further work is needed to explore how these design changes impact accuracy, robustness, and user perceptions.

[12] Awaiba NanEye GS Idule Demo Kit
[13] Awaiba NanEye 2D Sensor

6.5 Limitations

Our system design and studies had several limitations. The TouchCam camera needed to be manually focused and its relatively narrow field of view resulted in an offset between what the user was touching and what was sensed—the latter was particularly problematic for small locations (*e.g.*, finger tips). Future work should explore auto-focusing camera hardware with wide angle lens. The data collected during Studies I and II was collected under controlled conditions. Moreover, while the visually impaired participants in Study II were able to use TouchCam to complete all of the specified tasks, they occasionally needed multiple attempts to do so. Future work should explore more realistic and longitudinal usage.

7 Conclusion

We introduced and investigated TouchCam, a finger-worn, multi-sensor system that supports input at a variety of body locations while mitigating camera framing issues that blind users often experience. Our design also enables new types of contextual gestures based on location. We evaluated two iterations of the TouchCam system in terms of accuracy and robustness, as well as usability for our target group of visually impaired users. Our findings not only highlight the feasibility of our approach—greater than 95% accuracy at detecting 24 location-specific gestures, and support for realtime interaction at approximately 35 frames per second—but also characterize tradeoffs in robustness and usability between different types of on-body input. Fine-grained input on the palm and fingers is desirable for efficient and discrete input, but these locations are more challenging to classify reliably due to their small size and similar visual features; in contrast, disparate body locations are easier to recognize and may enable more intuitive mappings between location and application, but may also be less efficient for a new user and potentially socially unacceptable. Location-specific gestures have the potential to support efficient interaction for expert users, flexible input locations depending on user preference or situation (*e.g.*, while walking with a cane *vs.* sitting at home), task-based interactions tied to intuitive locations, and relatively fine-grained input for body areas that have distinctive visual features (*e.g.*, fingertips and palm). In future work, we plan to explore ways to improve robustness and evaluate our system's long-term performance during a longitudinal study.

8 Acknowledgements

This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs under Award W81XWH-14-1-0617. We thank Liang He for his work on the TouchCam video (see supplementary materials) and our participants.

Appendix

Localization Features		**Motion Features**	
IR	2 raw IR sensor readings	IR	70 features: 50 resampled points + 5 summary statistics × 4 windows
IMUs	4D orientation vector (quaternion) for each IMU	IMUs	639 features: 3 sensors × [3 axes × (50 resampled points + 5 summary statistics × 4 windows) + 3 correlation values]
Camera	LBP texture histogram with 1792 bins (14 patterns x 16 variances x 8 scales) 2D Gabor keypoints, variable number per image	Camera	140 features: 2 axes × (50 resampled points + 5 summary statistics × 4 windows)

Table 4. Summary of localization and motion features extracted from each sensor for TouchCam Offline (Study I).

Single Sensors	t_{23}	p	d
IR vs. W	-17.90	<0.001	-3.65
IR vs. F	-17.16	<0.001	-3.50
IR vs. C	-10.38	<0.001	-2.12
W vs. C	4.34	<0.001	0.89
F vs. C	3.63	0.005	0.74

Three Sensors	t_{23}	p	d
IR+F+W vs. IR+F+C	6.46	<0.001	1.32
IR+F+C vs. F+W+C	-4.71	<0.001	-0.96
IR+F+W vs. IR+W+C	3.64	0.003	0.74
IR+F+W vs. F+W+C	-3.17	0.016	-0.65

Two Sensors	t_{23}	p	d
IR+C vs. F+W	-8.85	<0.001	-1.81
IR+C vs. W+C	-7.46	<0.001	-1.52
IR+W vs. IR+C	7.39	<0.001	1.51
IR+F vs. IR+C	6.72	<0.001	1.37
IR+C vs. F+C	-6.59	<0.001	-1.35
F+W vs. F+C	6.20	<0.001	1.27
IR+F vs. F+W	-4.67	<0.001	-0.95
F+W vs. W+C	3.73	0.008	0.76
IR+W vs. F+W	-3.23	0.036	-0.66

Best Three (IR+F+W) vs. All Four Sensors	t_{23}	p	d
IR+F+W vs. All	-2.13	0.044	-0.44

Best Single (W) vs. Two Sensors	t_{23}	p	d
W vs. IR+C	5.73	<0.001	1.17
W vs. W+C	-4.81	<0.001	-0.98
W vs. IR+W	-4.81	<0.001	-0.98
W vs. F+W	-4.75	<0.001	-0.97

Best Two (F+W) vs. Three Sensors	t_{23}	p	d
F+W vs. IR+F+C	4.17	<0.001	0.85
F+W vs. IR+F+W	-2.99	0.020	-0.61
F+W vs. F+W+C	-2.67	0.027	-0.55
F+W vs. IR+W+C	-2.67	0.026	0.49

Table 5. Statistically significant comparisons between sensors used in Study I.

REFERENCES

[1] Dustin Adams, Lourdes Morales, and Sri Kurniawan. 2013. A qualitative study to support a blind photography mobile application. In *Proc. PETRA 2013*: 1–8. https://doi.org/10.1145/2504335.2504360

[2] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proc. CHI 2011*, 2043–2046. https://doi.org/10.1145/1978942.1979238

[3] John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 6: 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

[4] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proc. UIST 2015*, 549–556. https://doi.org/10.1145/2807442.2807450

[5] Liwei Chan, Chi-Hao Hsieh, Yi-Ling Chen, Shuo Yang, Da-Yuan Huang, Rong-Hao Liang, and Bing-Yu Chen. 2015. Cyclops: Wearable and single-piece full-body gesture input devices. In *Proc. CHI 2015*, 3001–3009. https://doi.org/10.1145/2702123.2702464

[6] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proc. UIST 2013*, 255–260. https://doi.org/10.1145/2501988.2502016

[7] Michał Choraś and Rafał Kozik. 2012. Contactless palmprint and knuckle biometrics for mobile devices. *Pattern Anal. Appl.* 15, 1: 73–85. https://doi.org/10.1007/s10044-011-0248-4

[8] Dar-Shyang Lee and S.N. Srihari. 1995. A theory of classifier combination: the neural network approach. In *Proc. ICDAR 1995*, 42–45. https://doi.org/10.1109/ICDAR.1995.598940

[9] Mohammad Omar Derawi, Bian Yang, and Christoph Busch. 2012. Fingerprint Recognition with Embedded Cameras on Mobile Phones. In *Security and Privacy in Mobile Info. and Com. Sys.* Springer, 136–147. https://doi.org/10.1007/978-3-642-30244-2_12

[10] Niloofar Dezfuli, Mohammadreza Khalilbeigi, Jochen Huber, Florian Müller, and Max Mühlhäuser. 2012. PalmRC: Imaginary Palm-Based Remote Control for Eyes-free Television Interaction. In *Proc. EuroITV 2012*, 27. Retrieved August 13, 2015 from http://dl.acm.org/citation.cfm?id=2325616.2325623

[11] Murat Ekinci and Murat Aykut. 2008. Palmprint Recognition by Applying Wavelet-Based Kernel PCA. *Computer Science and Technology* 23, 107: 851–861.

[12] Eryun Liu, A. K. Jain, and Jie Tian. 2013. A Coarse to Fine Minutiae-Based Latent Palmprint Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 10: 2307–2322. https://doi.org/10.1109/TPAMI.2013.39

[13] Leah Findlater, Lee Stearns, Ruofei Du, Uran Oh, David Ross, Rama Chellappa, and Jon Froehlich. 2015. Supporting Everyday Activities for Persons with Visual Impairments Through Computer Vision-Augmented Touch. In *Proc.* ASSETS 2015, 383–384. https://doi.org/10.1145/2700648.2811381

[14] Yoav Freund and Robert E. Schapire. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Proc. EuroCOLT 1995.* Springer, Berlin, Heidelberg, 23–37. https://doi.org/10.1007/3-540-59119-2_166

[15] Daniel Goldreich and Ingrid M. Kanics. 2003. Tactile Acuity is Enhanced in Blindness. *J. Neurosci.* 23, 8: 3439–3445. Retrieved November 24, 2013 from http://www.jneurosci.org/content/23/8/3439.abstract

[16] Zhenhua Guo, Lei Zhang, and David Zhang. 2010. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition* 43, 3: 706–719. https://doi.org/10.1016/j.patcog.2009.08.017

[17] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback. In *Proc. UIST 2010*, 3–12. https://doi.org/10.1145/1866029.1866033

[18] Sean G Gustafson, Bernhard Rabe, and Patrick M Baudisch. 2013. Understanding Palm-based Imaginary Interfaces: The Role of Visual and Tactile Cues when Browsing. In *Proc. CHI 2013*, 889–898. https://doi.org/10.1145/2470654.2466114

[19] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proc. UIST 2011,* 283–292. https://doi.org/10.1145/2047196.2047233

[20] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body As an Input Surface. In *Proc. CHI 2010*, 453–462. https://doi.org/10.1145/1753326.1753394

[21] Chris Harrison and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proc. UIST 2011*, 441–450.

[22] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*: 65–70.

[23] De-Shuang Huang, Wei Jia, and David Zhang. 2008. Palmprint verification based on principal lines. *Pattern Recognition* 41, 4: 1316–1328. https://doi.org/10.1016/j.patcog.2007.08.016

[24] A.K. Jain and Jianjiang Feng. 2009. Latent Palmprint Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 6: 1032–1047. https://doi.org/10.1109/TPAMI.2008.242

[25] Chajndrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P. Bigham. 2011. Supporting blind photography. In *Proc. ASSETS 2011*, 203. https://doi.org/10.1145/2049536.2049573

[26] Wolf Kienzle and Ken Hinckley. 2014. LightRing. In *Proc. UIST 2014*, 157–160. https://doi.org/10.1145/2642918.2647376

[27] Gierad Laput, Robert Xiao, Xiang "Anthony" Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin buttons. In *Proc. UIST 2014*, 389–394. https://doi.org/10.1145/2642918.2647356

[28] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers. In *Proc. UIST 2016*, 321–333. https://doi.org/10.1145/2984511.2984582

[29] John P. Lewis. 1995. Fast Template Matching. *Vision Interface* 95, 120123: 15–19.

[30] 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 on - SA '11*, 1–1. https://doi.org/10.1145/2073370.2073374

[31] Soo-Chul Lim, Jungsoon Shin, Seung-Chan Kim, and Joonah Park. 2015. Expansion of Smartwatch Touch Interface from Touchscreen to Around Device Interface Using Infrared Line Image Sensors. *Sensors* 15, 7: 16642–16653. https://doi.org/10.3390/s150716642

[32] 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 *Proc. UIST 2011*, 481–488. https://doi.org/10.1145/2047196.2047259

[33] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, 1–7. https://doi.org/10.1109/ICORR.2011.5975346

[34] Heinrich Braun Martin Riedmiller. 1992. RPROP - A Fast Adaptive Learning Algorithm. In *Proc. of ISCIS VII*.

[35] Denys J. C. Matthies, Simon T. Perrault, Bodo Urban, and Shengdong Zhao. 2015. Botential: Localizing On-Body Gestures by Measuring Electrical Signatures on the Human Skin. In *Proc, MobileHCI 2015*, 207–216. https://doi.org/10.1145/2785830.2785859

[36] Abdallah Meraoumia, Salim Chitroub, and Ahmed Bouridane. 2011. Fusion of Finger-Knuckle-Print and Palmprint for an Efficient Multi-Biometric System of Person Recognition. In *2011 IEEE International Conference on Communications (ICC)*, 1–5. https://doi.org/10.1109/icc.2011.5962661

[37] Aythami Morales, Miguel A. Ferrer, and Ajay Kumar. 2010. Improved palmprint authentication using contactless imaging. In *IEEE Conf. on Biometrics: Theory, Applications and Systems (BTAS)*, 1–6. https://doi.org/10.1109/BTAS.2010.5634472

[38] Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. 2013. EyeRing: A Finger-worn Input Device for Seamless Interactions with Our Surroundings. In *Proc. AH 2013*, 13–20. https://doi.org/10.1145/2459236.2459240

[39] J Farley Norman and Ashley N Bartholomew. 2011. Blindness enhances tactile acuity and haptic 3-D shape discrimination. *Attention, Perception, & Psychophysics* 73, 7: 2323–2331. https://doi.org/10.3758/s13414-011-0160-4

[40] Masa Ogata and Michita Imai. 2015. SkinWatch. In *Proc. AH 2015*, 21–24. https://doi.org/10.1145/2735711.2735830

[41] Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: Adapting Skin as a Soft Interface. In *Proc. UIST 2013*, 539–544. https://doi.org/10.1145/2501988.2502039

[42] Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2014. Augmenting a Wearable Display with Skin Surface as an Expanded Input Area. In *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, Aaron Marcus (ed.). Springer International Publishing, Cham, 606–614. https://doi.org/10.1007/978-3-319-07626-3_57

[43] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proc. UIST 2012*, 131–136.

[44] Uran Oh and Leah Findlater. 2014. Design of and subjective response to on-body input for people with visual impairments. In *Proc. ASSETS '14*, 8 pages.

[45] Uran Oh and Leah Findlater. 2015. A Performance Comparison of On-Hand versus On-Phone Non-Visual Input by Blind and Sighted Users. *ACM Transactions on Accessible Computing (TACCESS)* 7, 4: 14.

[46] Uran Oh, Lee Stearns, Alisha Pradhan, Jon E. Froehlich, and Leah Findlater. 2017. Investigating Microinteractions for People with Visual Impairments and the Potential Role of On-Body Interaction. In *Proc. ASSETS 2017*, TO APPEAR.

[47] John C. Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers* 10, 3: 61–74.

[48] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proc. CHI 2012*, c: 483–492. https://doi.org/10.1145/2207676.2207743

[49] Roy Shilkrot, Jochen Huber, Jürgen Steimle, Suranga Nanayakkara, and Pattie Maes. 2015. Digital Digits: A Comprehensive Survey of Finger Augmentation Devices. *ACM Computing Surveys* 48, 2: 1–29. https://doi.org/10.1145/2828993

[50] 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 *Proc. CHI 2017*, 3891–3902. https://doi.org/10.1145/3025453.3026005

[51] Lee Stearns, Ruofei Du, Uran Oh, Catherine Jou, Leah Findlater, David A. Ross, and Jon E. Froehlich. 2016. Evaluating Haptic and Auditory Directional Guidance to Assist Blind People in Reading Printed Text Using Finger-Mounted Cameras. *ACM Transactions on Accessible Computing* 9, 1: 1–38. https://doi.org/10.1145/2914793

[52] Lee Stearns, Ruofei Du, Uran Oh, Yumeng Wang, Rama Chellappa, Leah Findlater, and Jon E. Froehlich. 2014. The Design and Preliminary Evaluation of a Finger-Mounted Camera and Feedback System to Enable Reading of Printed Text for the Blind. *Workshop on Assistive Computer Vision and Robotics (ACVR'14) in Conjunction with the European Conference on Computer Vision (ECCV'14).* https://doi.org/10.1007/978-3-319-16199-0_43

[53] Lee Stearns, Uran Oh, Bridget J. Cheng, Leah Findlater, David Ross, Rama Chellappa, and Jon E. Froehlich. 2016. Localization of skin features on the hand and wrist from small image patches. In *Proc. ICPR 2016*, 1003–1010. https://doi.org/10.1109/ICPR.2016.7899767

[54] Emi Tamaki, Takashi Miyaki, and Jun Rekimoto. 2009. Brainy Hand: an earworn hand gesture interaction device. In *Proc. CHI EA 2009*, 4255. https://doi.org/10.1145/1520340.1520649

[55] Marynel Vázquez and Aaron Steinfeld. 2012. Helping visually impaired users properly aim a camera. In *Proc. ASSETS 2012*, 95. https://doi.org/10.1145/2384916.2384934

[56] Wai Kin Kong and D. Zhang. 2002. Palmprint texture analysis based on low-resolution images for personal authentication. In *Proc. Pattern Recognition '02*, 807–810. https://doi.org/10.1109/ICPR.2002.1048142

[57] Cheng-Yao Wang, Min-Chieh Hsiu, Po-Tsung Chiu, Chiao-Hui Chang, Liwei Chan, Bing-Yu Chen, and Mike Y. Chen. 2015. PalmGesture: Using Palms as Gesture Interfaces for Eyes-free Input. In *Proc. MobileHCI 2015*, 217–226. https://doi.org/10.1145/2785830.2785885

[58] 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 *Proc. CHI 2015*, 2991–3000. https://doi.org/10.1145/2702123.2702391

[59] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. 2009. Gesture Recognition with a 3-D Accelerometer. In *Ubiquitous intelligence and computing (UIC 2009)*, 25–38. https://doi.org/10.1007/978-3-642-02830-4_4

[60] Xiangqian Wu, Qiushi Zhao, and Wei Bu. 2014. A SIFT-based contactless palmprint verification approach using iterative RANSAC and local palmprint descriptors. *Pattern Recognition* 47, 10: 3314–3326. https://doi.org/10.1016/j.patcog.2014.04.008

[61] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proc. UIST 2012*, 147–156. https://doi.org/10.1145/2380116.2380137

[62] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Sking. In *Proc. CHI 2016*, 1491–1503. https://doi.org/10.1145/2858036.2858082

[63] 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 *Proc. CHI 2016*, 1491–1503. https://doi.org/10.1145/2858036.2858082

[64] Zhi Li, Guizhong Liu, Yang Yang, and Junyong You. 2012. Scale- and Rotation-Invariant Local Binary Pattern Using Scale-Adaptive Texton and Subuniform-Based Circular Shift. *IEEE Transactions on Image Processing* 21, 4: 2130–2140. https://doi.org/10.1109/TIP.2011.2173697

Received May 2017; revised August 2017; accepted October 2017