FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures

Mohamed Soliman[1,2], **Franziska Mueller**[1,2], **Lena Hegemann**[1,2],
Joan Sol Roo[1,2], **Christian Theobalt**[2], **Jürgen Steimle**[1,2]
[1]Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
[2]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
{soliman, roo, steimle}@cs.uni-saarland.de, {frmueller, theobalt}@mpi-inf.mpg.de

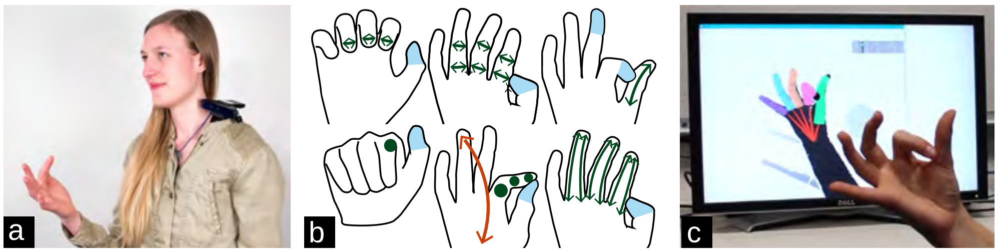

Figure 1. (a) FingerInput enables detection of versatile thumb-to-finger interactions using a body-worn depth camera. (b) By detecting finger flexion as well as touch locations, it supports a broad set of microgestures. (c) This is achieved thanks to a hybrid method that combines Convolutional Neural Networks, a fully articulated hand model, and sphere-based continuous collision detection.

ABSTRACT

Single-hand thumb-to-finger microgestures have shown great promise for expressive, fast and direct interactions. However, pioneering gesture recognition systems each focused on a particular subset of gestures. We are still in lack of systems that can detect the set of possible gestures to a fuller extent. In this paper, we present a consolidated design space for thumb-to-finger microgestures. Based on this design space, we present a thumb-to-finger gesture recognition system using depth sensing and convolutional neural networks. It is the first system that accurately detects the touch points between fingers as well as the finger flexion. As a result, it can detect a broader set of gestures than the existing alternatives, while also providing high-resolution information about the contact points. The system shows an average accuracy of 91% for the real-time detection of 8 demanding thumb-to-finger gesture classes. We demonstrate the potential of this technology via a set of example applications.

CCS Concepts

•**Human-centered computing** → **Human computer interaction;** *Interaction techniques; Interactive systems and tools;*

Author Keywords

Gestural input; microgesture; finger touch input; on-body input; skin input; depth sensor; gesture recognition.

ISS '18, November 25–28, 2018, Tokyo, Japan

ISBN 123-4567-24-567/08/06. . . $15.00
DOI: https://doi.org/10.1145/3279778.3279799

INTRODUCTION

There has been a growing interest in thumb-to-finger gestures, which take advantage of the inherent fine motor skills of fingers to allow users to expressively control digital systems [14, 2, 5, 19]. By touching one or multiple fingers with the thumb, the user can perform touch input directly on the skin. This promises to be a very direct, fast, and discreet type of input, even more as it supports one-handed and eyes-free interactions. However, sensing such thumb-to-finger gestures is hard because these gestures involve small movements and are performed at a body location that is difficult to instrument.

Pioneering research has demonstrated a variety of gestures and presented various approaches to sensing [14, 50]. Considering the recency of the field, it is not surprising that it is characterized by point explorations, focusing on a specific and rather small subset of thumb-to-finger gestures. As a result, these recognition systems are typically developed to demonstrate novel interactions and therefore limited to specific instances, such as tapping on a finger segment or sliding along a finger. While viable for the purpose, such restricted gesture sets limit the scope of possible mappings in real-world applications. So far, it remained unclear whether the conceptual space of possible gestures has been fully covered, what are common design dimensions, and—most important from a technical perspective—how the various gestures can be integrated in one system. We contribute to this emerging line of research by exploring how to support the set of thumb-to-finger gestures to a fuller extent.

In this work, we focus on supporting expressive, multidimensional thumb-to-finger interaction. To inform the design of gesture recognition systems for thumb-to-finger interactions, we first classify prior work and derive a consolidated design space using an open-coding approach, in which we identify gestural primitives of thumb-to-finger microgestures. The resulting design space provides a broader list of microgestures than previous work: the primitives cover existing gestures from the literature on hand-free microgrestures [14, 13, 44, 53] while also demonstrate opportunities for novel gestures. We use the design space to derive technical requirements for recognition systems that support a broader set of gestures.

We address these requirements by contributing a novel gesture recognition system for thumb-to-finger input. It is the first system that can capture all primitives of the design space and hence significantly extends the set of gestures that can be detected in an interactive system, adding to the expressiveness of input. The system is capable of identifying fingers and finger segments, tracking their 3D pose, and detecting linear and rotary touch contact between fingers, all with a high accuracy and in real time. Our system is vision-based and uses a body-worn depth sensor mounted on the user's head or shoulder. As such, it does not require any instrumentation of the hand, while not being affected by bad lighting as only depth information is used. We present three pilot studies to validate the functionality of the algorithm. Results from a fourth technical evaluation with users show a high accuracy of 91% on thumb-to-finger gesture recognition, for a rich variety of 8 gesture classes. Additionally, we demonstrate the practical feasibility with 2 example applications. They show the potential for expressive interactions that are enabled by our approach.

TECHNICAL RELATED WORK

The technical work presented in this paper builds on recent advances in body-based sensing and vision-based detection of body gestures:

Sensing On-Body Touch Input

Detecting touch input on the body has been approached using different sensing techniques, including acoustic sensing [12], inertial and magnetic sensing [5, 14, 6, 13], photo-reflective sensing [23], radar [48] or capacitive sensing [49, 50]. However, these approaches have some limitations: capacitive approaches have relatively low resolution, and the hand needs to be instrumented. Magnetic approaches have high resolution, but they do not provide accurate temporal touch detection. All the approaches are able to detect touch, but do not measure finger flexion for other fingers than the ones involved in the touch action [55, 54].

Another widely used approach consists of using a body-mounted camera. Possible camera locations include the head [7, 36, 9], shoulder [11, 52], chest [20, 4, 21], and wrist [33, 17, 26, 8]. OmniTouch [11] uses a depth camera and a projector mounted on the shoulder to turn the inside of the palm into a touch surface. Sridhar et al. [33] use a depth camera mounted on the wrist to enable 3D input on the back of the hand. PinchWatch [20] allows microinteractions by mounting a depth camera on the chest, which tracks the hand wearing a display. These approaches do not require instrumentation of the hand and generally tend to work robustly for larger touch or free-hand gestures that are based on the hand shape. However, it is a hard problem to accurately detect touch contact between fingers from a distant camera.

Finer finger gestures are addressed by Cyclopsring [3] by mounting a fish-eye camera on a ring, which is used to detect different touch gestures based on the hand shape. For detecting touch, some approaches use image based techniques, such as flood filling [11, 33]. While flood filling is suitable for detecting touch on a constrained touch area like a flat surface [11], or the back of the hand [33], it is not suitable for a wider class of touch interactions on more general and complex surfaces, like different finger segments. Even when geometric shapes have been used for hand tracking [27], we are not aware of prior work that used this approach for detecting touch points.

Recognizing Hand Pose

Another stream of research investigates capturing the detailed hand posture. To provide a flexible and lightweight setup, recent hand tracking algorithms tend to use a single consumer depth sensor. They can be divided into three classes: *discriminative*, *generative* and *hybrid*. *Discriminative* approaches are based on data-driven machine learning techniques. Recently, convolutional neural networks (CNNs) have been successfully used for hand pose estimation [40, 47, 32]. *Generative* approaches use a generative hand model for comparing the current pose estimate and the observation [39]. *Hybrid* methods combine discriminative and generative approaches to achieve both robust and accurate hand pose estimation and hand tracking [34, 31, 37, 45]. While most of the previous work focused on a fixed exocentric depth camera —which is impractical for mobile scenarios— only few explored egocentric settings with body-mounted cameras [28, 22, 38]. Although all hand tracking approaches estimate flexion angles for all fingers and some run in real-time, no method for precise continuous touch point estimation has been proposed.

Extending beyond prior work, our system is able to detect fine finger-to-finger interaction, involving the continuous rotational angle and relative position along the touched finger segment and an accurate detection of touch contact. Moreover, it provides continuous estimation of the flexion angles of all fingers and exact relative finger positions in 3D. As explained in the next section, this is sufficient for accurate detection of thumb-to-finger microgestures.

DESIGN SPACE OF THUMB-TO-FINGER GESTURES

In order to inform the design of recognition systems for thumb-to-finger microgestures, we extracted common features of finger articulation from the different gestures proposed in prior work. Using an open-coding approach, we identified gestural primitives and consolidated them into a four-dimensional design space. Together, these four dimensions define a thumb-to-finger gesture by: a) which finger is touching, b) what location on another finger is touched, c) what touch action is performed, and d) how fingers are flexed. The dimensions and their possible values are illustrated in Figure 2.

a. Touch Initiator

The first defining factor for microgestures is the finger that triggers the touch action, that is, either the thumb or one of the remaining fingers.

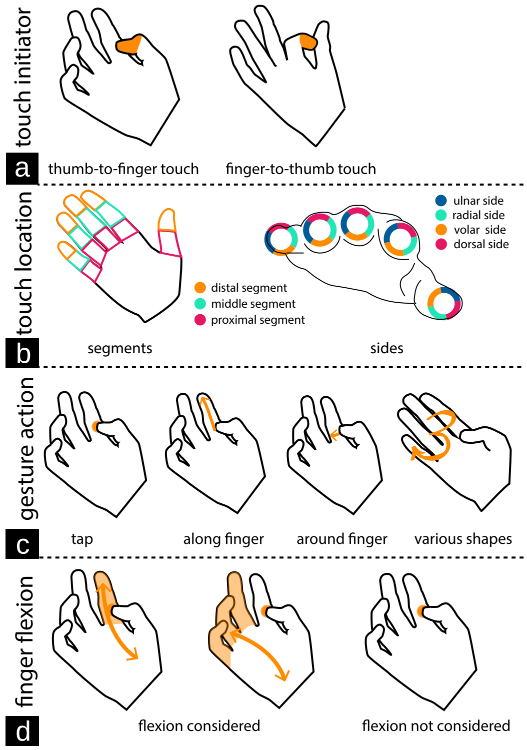

Figure 2. Thumb-to-finger gestures are defined by four dimensions: (a) The finger initiating the touch, (b) the touch location, (c) the gesture action, and (d) finger flexion.

Thumb-to-finger: The vast majority of prior work focused on gestures that are initiated by the thumb, which is touching another finger. Work investigating such thumb-to-finger input includes [5, 43, 42, 57, 50, 2, 14, 51]. This space is comparably well-covered, including systematic empirical studies that investigated the comfort regions for touch interactions initiated by the thumb [14].

Finger-to-thumb: The reverse interaction of a finger initiating a touch *on* the thumb has rarely been investigated. We call this *finger-to-thumb* input. This form of input has been implicitly used to extend the input area of finger sliding gestures [20] or to enable sliding gestures while holding an object [53]. So far, this form of input has been limited to linear sliding with the index finger on the thumb. There is the opportunity to extend to a wider variety of gestures that include tapping, rotational sliding, and use of other fingers.

We explicitly exclude a third class, which would include touch contact between any two fingers other than the thumb, as it lays out of the scope of thumb-and-finger interaction. Such touch contact is particularly challenging to detect correctly, as the contact is frequent and the involved areas are not only points but whole surfaces at the time.

b. Touch Location

The touched location is defined by the touched finger segment (*proximal*, *middle* or *distal*) and the rotary side of the finger (*radial*, *ulnar*, *dorsal* or *volar*). A tapping gesture contains only one touch location, while a continuous gesture contains multiple sequential locations.

Finger segment: Each finger can be divided into two segments (thumb) or three segments (other fingers). Because of the tactile and visual cues generated by knuckles and wrinkles, each segment is clearly delimited. This makes them a natural choice as touch targets, as seen in prior studies [42, 26, 51]. Some work has subdivided the segments even further [14, 43].

Finger side: The location of touch input on a finger is also defined by the rotary angle around the finger's longitudinal axis. The vast majority of work has focused on input performed on only one side of the finger: either the radial side [54, 43, 48] (i.e., the side closer to the thumb) or the volar (i.e., palmar) side [14, 30, 46]. Only very few studies have investigated input on other sides. Notable exceptions include work that investigates the likability of touch input on two sides of the fingers (radial, palmar) [42]. Other work applied tap on two sides of one segment [15] and pioneered sliding input around the finger segment [41, 24], finger nail [1], or finger wrinkles [50]. Overall, gestures that involve the different rotary sides of the fingers are still underexplored and present new opportunities for interaction.

c. Gesture Action

The gesture action defines what form of touch input the user is performing: either a tap, a continuous longitudinal or rotary sliding movement, or a specific shape that is drawn with the touch initiator.

Tapping: The touch initiator is touching a finger at a discrete location. The touch locations explored include the different segments and sides of the fingers [14, 54, 42, 5, 41, 48, 15, 10], as well as the fingernails [1].

Sliding along the finger: The touch initiator slides along the touched finger's longitudinal axis. The slide can be performed along the entire finger [57, 3, 51, 55, 53, 20, 10], or on a segment of the finger [50, 5, 41, 1, 30]. This set of actions is typically used to manipulate continuous values [51, 20, 3], but it has also been used for discrete gestures [50, 41, 30].

Sliding around the finger: The touch initiator slides perpendicular to the lateral axis of the touched finger [5, 1, 24, 41, 50]. The action can be also be performed on multiple fingers [53, 2, 57, 10].

Drawing shapes on the fingers: The initiator is used to draw a shape on one or more fingers. The action is completed once the shape is fully drawn and the touch contact released. Different shapes have been investigated, including circles [20], characters [14], and digits [57]. The drawing action can be performed on a single segment of one finger [30, 14], or on multiple fingers [2].

d. Finger Flexion

The flexion of the different fingers of the hand can be considered as a part of the performed gesture. This property adds an additional dimension to the touch gesture performed on the fingers. Each finger can be *open*, *folded* or *moving*. We use the terms open and folded similar to Krupka et al. [18] who defined a finger or the thumb as folded when its tip resides in a certain area in front of the palm. Finger flexion can be a discrete property. It can also be a continuous feature when one or more fingers are moving from open to folded or folded to open state during a gesture. We are aware of only one prior study that included finger flexion of the touched finger to execute different actions [54]. Combining thumb-to-finger touch with the expressive capabilities of free-hand gestures [48] opens up a promising direction for novel gestures.

Resulting Technical Requirements

Accurate detection of these gestural primitives of all four dimensions poses a set of demanding technical requirements for gesture recognition systems. We identify the following five main requirements:

1. Hand and finger segmentation
2. Identification of the touch initiator and of the touched finger
3. Estimation of the touch location, including linear (segment) and rotational (finger side) position
4. Temporal detection of touch contact (touch down vs. touch up), to identify the onset and offset of a gesture
5. Estimation of the flexion angles of all fingers
6. Real-time performance.

GESTURE RECOGNITION SYSTEM

We now describe our depth camera-based gesture recognition system for supporting versatile and expressive thumb-to-finger interactions. Our approach works in real time, with a single body-mounted depth sensor, and is able to reconstruct fine-grained thumb-to-finger interactions with high accuracy without requiring any instrumentation of the hand. Our approach combines the real-time reconstruction of a fully-articulated hand pose (based on a fully convolutional neural network, a kinematic skeleton and a Gaussian mixture model) with real-time detection of thumb-to-finger touch contact to accurately classify input gestures.

To capture input gestures, the user mounts a depth camera on the head or either shoulder, as shown in Figure 3. This placement follows strategies from prior work and ensures compatibility with AR/VR devices. For instance, future head-mounted displays will likely include a forward-facing depth camera. We use an Intel RealSense SR300 camera [1] with a sensing range of 20 cm to 120 cm to capture the depth images.

Algorithm Description

Figure 4 depicts an overview of our algorithm. Figure 5 shows examples of different touch poses processed by the algorithm pipeline. The algorithm consists of four main steps:

[1]https://software.intel.com/en-us/realsense/sr300camera

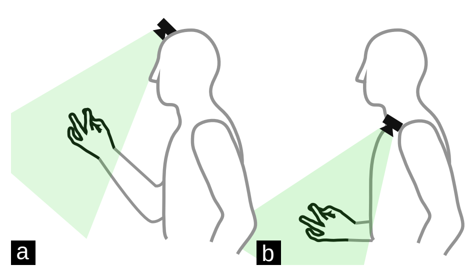

Figure 3. Thumb-to-finger touch gestures are captured using a depth camera that can be mounted on (a) the head or (b) the shoulder.

Hand part classification

We use a per-pixel classifier to segment the hand from the background and the arm, and to identify different fingers (Figure 4 (b)). Our classifier is a fully convolutional neural network inspired by the U-Net architecture [29]. The encoder part transforms the input depth image at resolution 240×320 to 256 feature maps of size 15×20 using 5 convolution layers and 1 max-pooling layer. Afterwards, the decoder generates a class label image at the original image resolution using 4 deconvolution layers. During training, the CNN tries to minimize the infogain loss where we set the weights according to the class frequencies to handle class imbalance. For training the CNN, we use the Caffe framework [16]. We train for 175,000 iterations with an input batch size of 16 using the AdaDelta [56] solver with momentum 0.95 and 0.0005. The base learning rate is lowered from 1.0 to 0.1 after 110,000 iterations.

To train our classifier, we collected real training data with automatic color-based labeling from a body-mounted RGBD camera at resolution 480×640. We collected data from 5 participants (3 males, 2 females; 22–27 years old) with the camera mounted at two different locations (head and shoulder), to provide an egocentric view of the hand for eyes-free input (Figure 3). To collect ground-truth information, we color-coded fingers, palm, back-of-the-hand, and arm, using finger paint, which is not visible in the IR image. This is the only step that required the color channels (RGB). We obtain per-pixel hand part labels for the depth image by HSV color segmentation. In total, 66,662 images were collected and automatically annotated. Of this set, 60,000 images were used for training the classifier. The remaining images were held out as a test set for classifier evaluation.

Hand pose and fingertip estimation

Recognizing finger-to-finger touch is a harder problem than identifying touch between a finger and a constrained, planar surface [11, 33]. It requires not only knowledge of fingertip positions but of the location of all bones in every finger. This full articulation of the hand is commonly described by 26 degrees of freedom (DOFs) [22]. For estimating all these DOFs, we

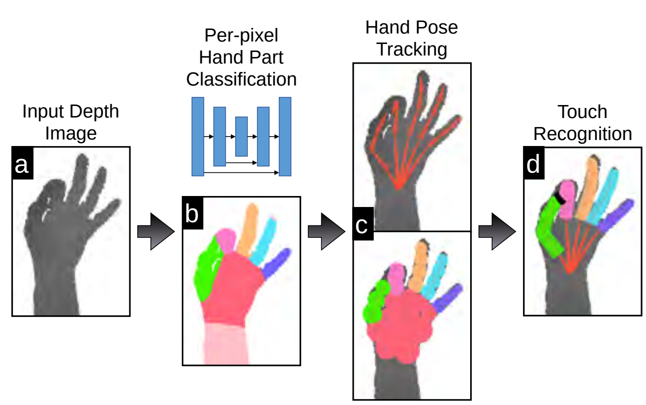

Figure 4. Algorithm processing steps: (a) Input depth image. (b) Using a CNN classifier, we obtain per-pixel hand part labels. (c) We estimate the full pose of the hand using a coarse generative model. (d) We attach touch proxies –which approximate the surface more accurately– to the kinematic skeleton and use them for continuous 3D touch recognition.

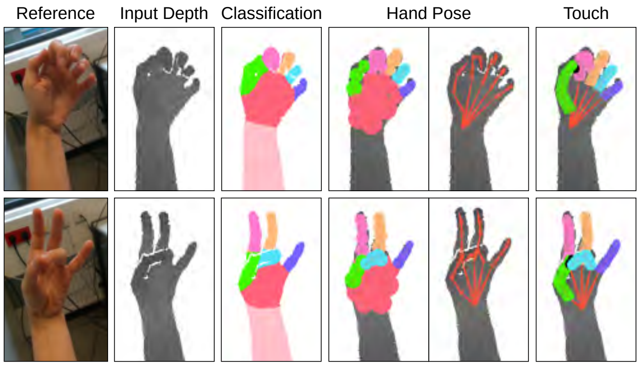

Figure 5. Different touch poses through the system pipeline. Touch position is indicated by a black circle.

use a generative model consisting of a kinematic skeleton and a 3D Gaussian mixture model (GMM). The GMM coarsely models the volumetric extent of the hand (see Figure 4 (c)) and has been successfully employed for full body as well as hand motion tracking [35, 34].

For every input depth image, we find the hand pose as the minimizer of an energy that describes the discrepancy of the generative model and the input observation. This energy function takes into account the per-pixel hand part labels obtained in the previous step and compares it to the pre-defined labels of the GMM. Furthermore, limits of joint angles and temporal smoothness are incorporated into the energy to ensure physical plausibility. This objective function is fast to optimize and has shown accurate and temporally stable results [34]. The fingertip positions and 3D location of every bone in the hand can be read from the posed kinematic skeleton after optimization.

Touch detection

To reliably recognize finger-to-finger touch contact based on the hand pose, we attach touch proxies to the hand model (see Figure 4 (d)). These approximate the surface more accurately than the isospheres of the Gaussians. The three segments of each finger are modeled as elliptical cylinders, the fingertips are modeled as spheres. These proxies are attached to the bones of the kinematic model so that they can easily be moved according to the tracked hand pose. Calculating a 3D touch point p can hence be formulated as sphere/cylinder and sphere/sphere intersection tests. The touch point p is then transformed to the local coordinate system (LCS) of the touched segment: $p_{loc} = T \cdot p$, where T is the transformation matrix from the global to the local coordinate system. Note that the LCS of a segment is setup s.t. the y-axis aligns with the bone and the x-z-plane is perpendicular to the bone. Thus, the exact relative geometric location of the touch point within the segment can be recovered: (1) the rotational angle around the bone is determined by the position of the touch point in the x-z-plane of the LCS, i.e. the x- and z-coordinate of p_{loc}, and (2) the position along the bone is derived from the y-coordinate of p_{loc}. The use of geometric primitives for touch recognition is computationally more efficient than other surface representations (e.g. meshes) while still allowing to model surfaces that are more complex than a single planar object.

Gesture classification

The touch information and the estimated hand pose are fed into a continuous gesture classifier to enable rich user interactions. The ability of FingerInput to detect exact touch instances provides a straightforward approach to recognizing gestures using voting-based discriminative classification in a continuous manner. Since all gestures involve touch contact, the classifier is activated when a touch instance starts and runs as long as it is present. The definition of gestures through the dimensions of the design space presented above allows for representing and defining gestures as a combination of values of these dimensions. A dictionary of gestures, defined by the values of the given dimensions, is stored in the system. For recognition, and on each frame, the values of each of the four dimensions are observed: touch initiator, touch location, gesture action, and finger flexion. A vote is then added to the corresponding combination of the dimensions, which maps to one of the defined gestures in the dictionary. We then use a time sliding window across frames to determine the performed gesture through majority voting, where the gesture with the most votes in the current window is selected. Empirical results, presented in the system evaluation section below, demonstrate that the system is able to detect a versatile set of gestures with a high accuracy.

Example Gesture Set

We selected a representative set of demanding microgestures (Figure 6). This selection includes discrete tapping as well as continuous movement along and around fingers, and also circle shapes. It further includes gesturing at many different finger locations, using the thumb or the index finger as initiator, and includes finger flexion. We decided to exclude interactions on and with the pinkie finger, since the largest part of the pinkie finger lays outside the comfort region of interaction [14].

To our knowledge, this set of gestures is considerably more versatile than the thumb-to-finger microgestures presented in any single prior system. Note that our system is not restricted to those gestures, but can be easily trained to detect other gestures that are based on the dimensions of the design space.

The gesture set is comprised of the following:

Finger tap: For this set of gestures, we evaluate taps on 9 segments on index, middle and ring finger. These gestures, known from prior work, are quick and easy to perform.

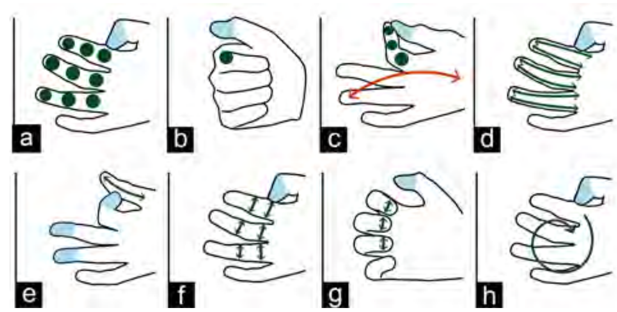

Figure 6. The 8 different classes of the evaluation gesture set: (a) finger taps, (b) fist tap, (c) tap-and-flap, (d) linear thumb-to-finger slides, (e) linear finger-to-thumb slides, (f) rotational thumb-to-finger slides, (g) fingertip slides, and (h) drawing a circle.

Fist tap: The hand forms a fist, all fingers being folded, while the thumb taps on a segment on the outer side of the index finger.

Tap-and-flap: Tapping based input can also be combined with dynamic finger poses. The thumb is tapping on the outer side of one of the index finger segments, while the other fingers move either from open to folded or vice versa.

Linear thumb-to-finger slide: The thumb is the touch initiator and performs a linear slide along the index, middle or ring finger. We include sliding on the inner and outer side of each finger, as well as in both directions, sliding from the root of the finger to the tip and vice versa, resulting in 12 gestures.

Linear finger-to-thumb slide: A similar sliding gesture can also be performed by the index or middle finger on the thumb.

Rotational thumb-to-finger slide: The thumb is the touch initiator and performs a rotational slide around one of the 6 inner and middle finger segments of the index, middle and ring finger while all fingers are open. We include sliding in both directions, resulting in overall 18 gestures.

Fingertip slide: The thumb can perform a rotational slide around the fingertip following the curve of the fingernail. This gesture applies to the index, middle, and ring fingers and can be performed in both directions.

Draw circle: The thumb starts from the lower segment of the index and draws a circle on the fingers.

Prototype System

Our prototype system runs at 40 frames per second on a desktop computer, using a Nvidia GTX 1080 GPU. The system recognizes gestures performed and sends events to clients through a WebSocket connection. Clients such as smart watches, smart phones, wall displays, or head-mounted displays can receive this information over a wireless connection.

SYSTEM EVALUATION

The objective of our contribution is to provide a tracker that accurately detects an expressive set of thumb-to-finger microgestures. To validate this claim, we first conducted a series of pilot studies to assess the correct and accurate functioning of the individual processing steps of the algorithm. Our main evaluation study then investigated the end-to-end functionality of the approach by assessing the system's accuracy to detect demanding microgestures that were performed by users. The findings confirm the functionality and accuracy of our system.

Pilot Study 1: Finger Classification

The first requirement for gesture detection is the correct segmentation of regions in the camera image. As described in the **Implementation** Section, we collected 66,662 depth images and automatically annotated them per-pixel for 6 semantic hand part classes and a non-hand class. 60,000 of these images were used for classifier training and 6,662 images were held out as test set. Table 1 shows the classification accuracies per class. Our CNN classifier achieves an average accuracy of **90.2%**, which shows its ability to accurately classify different fingers and to detect the exact fingers involved in a gesture.

Region	Palm	Thumb	Index	Middle	Ring	Pinkie	Non-hand
Accuracy	96.0%	93.5%	91.6%	87.3%	79.1%	84.8%	99.0%

Table 1. Per-class accuracy achieved by our classifier for the classes.

Pilot Study 2: 3D Fingertip Localization

To characterize the system's accuracy in detecting the positions of the fingertips in 3D space, two participants (1 female, 1 male; 23 and 27 years old) with average hand size (hand lengths: 201 mm, 192 mm; hand widths: 90 mm, 89 mm) [25] interacted with the system by performing gestures from the gesture dataset (Figure 6) as indicated by a video reference.

The collected dataset (dataset: 3,573 frames) was used to train our proposed CNN-based tracking system, and as an egocentric RDF-based tracker [34] as a baseline reference. For ground truth, we annotated the fingertip positions in each of the depth images. Then we calculated the average fingertip localization error by computing the Euclidean error of the 5 fingertip positions averaged over all frames.

Our proposed tracking system outperforms the egocentric RDF-based tracker by a large margin (**13.92 mm** and **16.37 mm** vs. 22.5 mm and 38.0 mm). The results show that the system has the capability to localize fingertips of different fingers with an error less than the average finger segment size. Note that the average error of the egocentric RDF-based tracker is higher than the error achieved by the corresponding third person tracker proposed in [34] on common third person datasets. This seems to indicate that our egocentric sequences are more challenging than third person settings due to frequent self-occlusion of the hand and additional camera motion.

Pilot Study 3: Touch Contact Between Fingers

To measure touch accuracy, we used the second dataset to compare the tracker output with ground truth data acquired via self-capacitive touch sensing (Figure 7). Capacitive touch is very accurate as it does not confuse finger touch down events with finger hover state and it is unaffected by occlusions. To capture capacitive ground-truth data, we custom-built a very thin and flexible sensor that participants wore on the thumb during this pilot study (see Figure 7). Two participants (1 female, 1 male; 23 and 27 years old) performed the same gestures as in Pilot Study 2. This dataset (5,174 frames) was automatically annotated with ground-truth information.

Figure 7. Automatic ground-truth annotation of thumb-to-finger touch using a capacitive finger glove, built out of nitrile rubber with conductive fabric affixed around the tip of the finger.

We calculated the accuracy using a frame-by-frame comparison as well as a time window comparison. For the frame-by-frame comparison, we calculate the number of frames where a touch is detected both by the system and in the ground truth data, with an average accuracy of **87.5%**. Most of the misclassified frames lie either directly before or after a performed touch gesture. These one-frame errors can be addressed with a time window: at a window length of 500 ms, the system had an average accuracy of **95.8%**. Lengths of 600 and 700 ms resulted in **96.2%** and **97.1%**, respectively.

Main Evaluation Study: Gesture Detection Accuracy

Methodology

To formally evaluate the accuracy of end-to-end gesture recognition, we performed a study with users. We recruited 10 volunteers (6 males, 4 females, 19–29 years old). Their hand lengths varied from 155–212 mm, and the hand widths from 74–92 mm (mean dimensions were 180 mm by 86 mm), providing a representative sample of hand size distribution [25]. The task consisted of performing all gestures in the 8 gesture classes presented in Fig. 6, with 4 trials each. Per gesture, two trials were performed while sitting, and the two others in a standing position. For each trial, participants started with an open-hand pose with no touching fingers, and then performed the touch gesture shown on a screen. Participants were instructed on how to perform the gestures, and were given several minutes to practice the gestures until they were comfortable with them. Gestures were recorded with an Intel RealSense SR300 depth camera, mounted on the participant's shoulder using a shoulder mount. Each recording session lasted 1 hour, with a break after performing 25 gestures. Our classifier evaluated the performed gestures for each frame.

Results

The overall accuracy of the system for gesture classification was **91.06%**. The confusion matrix is shown in Figure 8. As the results reveal, the fist tap and the circle drawing has the highest accuracy of 91.8%. This validates the ability of our system to avoid false activations of primary actions. Linear thumb-to-finger slides have the lowest accuracy of 89.9%. Rotational slides are at times confused with finger taps, as they both occur inside one segment.

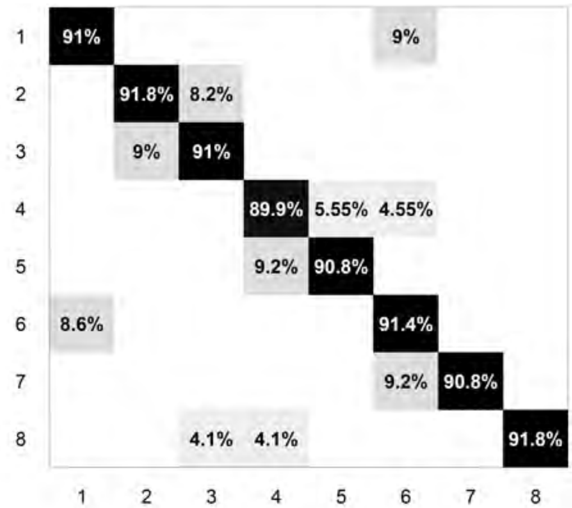

Figure 8. Confusion matrix for the 8 different classes of our gesture set. The 8 classes are (1) finger taps, (2) fist tap, (3) tap-and-flap, (4) linear thumb-to-finger slides, (5) linear finger-to-thumb slides, (6) rotational thumb-to-finger slides, (7) fingertip slides, and (8) drawing a circle. Our system has an accuracy of 91.06% in classifying the gestures.

EXAMPLE APPLICATIONS

In order to explore the potential of expressive microgestures, we implemented two example applications. As showcase scenario, we selected the interaction with audio data, as it is well-compatible with eyes-free interaction and requires different degrees of expressivity: from rapid and simple controls to expressive authoring and parametrization. Each of these examples integrates a wider variety of thumb-to-finger microgestures than in prior work. All gestures are captured in real-time using our tracker.

Direct and Rapid Control of a Music Player

Controlling a music player application benefits from easy, direct and rapid commands that are performed as one-handed and eyes-free input (Figure 9 (a)). While the few most important commands (play, pause, previous/next song) should be most rapidly available, music playback requires a larger set of secondary commands (e.g., control volume, control sound properties). At the same time, gestures should be chosen to minimize the likelihood of false activation, given the audio player is likely to be controlled during another primary activity. We address those conflicting goals by combining diverse thumb-to-finger microgestures with specific hand poses.

The application is invoked by performing a fist gesture for half a second, *grabbing* control of the application while avoiding false positives. In this hand pose, the radial side of the index finger forms a round surface that affords ergonomic and rapid tapping and sliding. In order to play/pause the current song, the user taps with the thumb on the middle segment of the index finger. The previous and next song can be selected by swiping left or right towards the distal or proximal segments respectively.

Stretching out the index finger serves as a fast and easy way to activate the mode for audio adjustments while avoiding false positives. The straight finger affords linear sliding along all three segments on the palmar side of the index finger. This allows for quick and precise adjustment of the volume. In turn, rotary slides around the distal, middle and proximal phalanx (finger segment) allow to equalize the bass, middle, and treble of the tone using microgestures. The use of this varied set of microgestures alongside specific hand poses has not been possible with prior systems.

Expressive Digital Music Composition

Historically, musical instruments have utilized the dexterity and expressiveness of the human hand and fingers in a variety of ways, combining discrete notes with continuous fine adjustments. We used this commonly performed practice as a source of inspiration to demonstrate the feasibility of thumb-to-finger gestures in an application for composing digital music using a sequencer (Figure 9 (b)).

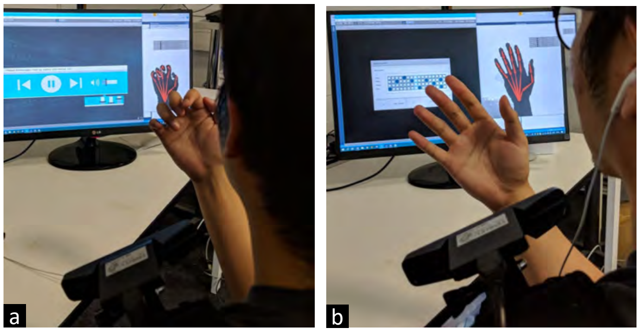

Figure 9. Audio control using microgestures. (a) Direct and rapid playback control. (b) Music composition using microgestures.

Our application is controlled solely by previously presented microgestures. The three independent channels of the sequencer are mapped to the three fingers - index, middle, and ring (pinky excluded). The 3 channels can be controlled simultaneously, allowing multidimensional control.

The instrument assigned to a finger can be selected via swiping on the distal-volar segment of that finger. Tapping the fingertip will trigger a single note from the selected instrument. Two properties can be controlled per channel: (1) volume control by linear sliding on the volar side of the finger, (2) pitch is controlled by sliding on the ulnar side.

Playback is controlled similar to the previous application, by tapping on the radial side of the index finger. While playing, performing the aforementioned operations will record the changes for the current beat. The proposed interaction combines commonly used tapping technique of creating rhythmic compositions, with expressive finger sliding to control the sound texture; this is demonstrated by the tracker capabilities to detect touch over different hand surfaces.

DISCUSSION AND LIMITATIONS

Selected gesture set: The presented gesture set is a selected subset from the possibilities present in the design space. These gestures were chosen to highlight the flexibility of the tracker with an heterogeneous gesture set, yet there are in no way exhaustive. Given that the tracker accurately recognizes the hand pose and touch points, other microgestures could be easily supported; a notable example of this would be to take into consideration the hand position and orientation, which are computed as part of the tracking process.

Occlusions: The results from our evaluation show that FingerInput achieves thumb-to-finger gesture classification with low false positives and can detect the fingertip location and classify fingers accurately. However, since our technique is based on an optical approach, it is limited by line-of-sight requirements: some occlusions resulting from crossed fingers, overly bent fingers, or rotated hands can be problematic for gesture classification. For this reason, our current setup assumes the user is holding the hand in a relaxed pose in front of the chest.

Hand model: The current version of the system requires that the measurements of the hand are manually indicated once for each user. Future implementations will include an initialization step.

Misclassification: We noticed that tap gestures can be misclassified as rotational slides when the user is slightly moving the finger while tapping. While this demonstrates the high spatial resolution of rotational slide, which even detects minimal movement, it may result in an undesired command. A straightforward solution consists of increasing the threshold for a minimal sliding movement. The trade-off between spatial resolution of rotational slides and robust detection of finger taps is an important question that should be further investigated in future work. An alternative design solution consists of spatial multiplexing: Drawing from the many different finger locations supported by our system, the designer can reserve some segments or finger sides for tap gestures, while others are used for sliding input.

Mobility: Our current prototype uses a desktop computer. The mobility could be extended by connecting the depth camera to a body-worn microcontroller (e.g., Raspberry Pi), and streaming depth data to a server for gesture classification.

CONCLUSION

We have presented a consolidated interaction design space for thumb-to-finger gestures and derived requirements for gesture recognition systems. Informed by these findings, we developed FingerInput, a system for versatile thumb-to-finger touch gestures, using a depth sensor. Our system covers the dimensions of the design space, and recognizes discrete and continuous thumb-to-finger touch gestures. Our results demonstrate that it is possible to implement a gesture recognition system that considerably extends the types of gestures that are supported in a single interface. On a conceptual level, our findings did not only allow us to classify existing thumb-to-finger gestures, but also helped us in identifying various opportunities for novel and extended gestures. Together, they show that thumb-to-finger gestures are rich and versatile and offer strong support for direct and single-handed interaction.

ACKNOWLEDGMENTS

This work was supported by the ERC Starting Grant Interactive Skin (714797), the ERC Starting Grant CapReal (335545), and the Cluster of Excellence on Multimodal Computing and Interaction. We thank Alyona Morozova for her help with the experiments, Xi Chen and Adwait Sharma for their help with applications and video editing.

REFERENCES

1. 2015. NailO: Fingernails as an Input Surface. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (2015), 3015–3018. DOI: http://dx.doi.org/10.1145/2702123.2702572

2. Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3403–3414.

3. Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015a. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 549–556.
4. Liwei Chan, Chi-Hao Hsieh, Yi-Ling Chen, Shuo Yang, Da-Yuan Huang, Rong-Hao Liang, and Bing-Yu Chen. 2015b. Cyclops: Wearable and single-piece full-body gesture input devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3001–3009.
5. 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. *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13* (2013), 255–260. DOI: http://dx.doi.org/10.1145/2501988.2502016
6. Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1504–1514. DOI: http://dx.doi.org/10.1145/2858036.2858125
7. Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K Goyal. 2013. Mime: compact, low power 3D gesture sensing for interaction with head mounted displays. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 227–236.
8. Artem Dementyev and Joseph A Paradiso. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 161–166.
9. Markus Funk, Sven Mayer, Michael Nistor, and Albrecht Schmidt. 2016. Mobile In-Situ Pick-by-Vision: Order Picking Support Using a Projector Helmet. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '16)*. ACM, New York, NY, USA, Article 45, 4 pages. DOI: http://dx.doi.org/10.1145/2910674.2910730
10. 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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 889–898.
11. Chris Harrison, Hrvoje Benko, and Andrew D Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 441–450.
12. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 453–462.
13. Christopher-Eyk Hrabia, Katrin Wolf, and Mathias Wilhelm. 2013. Whole hand modeling using 8 wearable sensors: biomechanics for hand pose prediction. In *Proceedings of the 4th Augmented Human International Conference*. ACM, 21–28.
14. Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1526–1537.
15. Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, Min-Lun Tsai, Yen-Ting Yeh, Liwei Chan, Yi-Ping Hung, and Mike Y Chen. 2014. TouchSense: expanding touchscreen input vocabulary using different areas of users' finger pads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 189–192.
16. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 675–678.
17. David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 167–176.
18. Eyal Krupka, Kfir Karmon, Noam Bloom, Daniel Freedman, Ilya Gurvich, Aviv Hurvitz, Ido Leichter, Yoni Smolin, Yuval Tzairi, Alon Vinnikov, and Aharon Bar Hillel. 2017. Toward Realistic Hands Gesture Interface: Keeping it Simple for Developers and Machines. In *Proceedings of ACM CHI*. 14. https://sites.google.com/site/aharonbarhillel/CHI_2017.pdf?attredirects=0
19. Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 142.
20. Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: a wearable device for one-handed microinteractions. In *Proc. MobileHCI*, Vol. 10.
21. Pranav Mistry, Pattie Maes, and Liyan Chang. 2009. WUW-wear Ur world: a wearable gestural interface. In *CHI'09 extended abstracts on Human factors in computing systems*. ACM, 4111–4116.

22. Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-time Hand Tracking under Occlusion from an Egocentric RGB-D Sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*. `http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/`
23. Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: adapting skin as a soft interface. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 539–544.
24. Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: Intelligent Ring Using Infrared Reflection. *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12* (2012), 131–136. DOI: `http://dx.doi.org/10.1145/2380116.2380135`
25. Alan Poston. 2000. Human engineering design data digest. *Washington, DC: Department of Defense Human Factors Engineering Technical Advisory Group* (2000).
26. Manuel Prätorius, Dimitar Valkov, Ulrich Burgbacher, and Klaus Hinrichs. 2014. DigiTap: an eyes-free VR/AR symbolic input device. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*. ACM, 9–18.
27. Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1106–1113.
28. Grégory Rogez, Maryam Khademi, JS Supančič III, Jose Maria Martinez Montiel, and Deva Ramanan. 2014. 3D hand pose detection in egocentric RGB-D images. In *Workshop at the European Conference on Computer Vision*. Springer, 356–371.
29. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR* abs/1505.04597 (2015). `http://arxiv.org/abs/1505.04597`
30. Kyeongeun Seo and Hyeonjoong Cho. 2014. AirPincher: a handheld device for recognizing delicate mid-air hand gestures. *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology - UIST'14 Adjunct* (2014), 83–84. DOI: `http://dx.doi.org/10.1145/2658779.2658787`
31. Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, and others. 2015. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3633–3642.
32. Ayan Sinha, Chiho Choi, and Karthik Ramani. 2016. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4150–4158.
33. Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor. In *Proceedings of ACM CHI*. 12. `http://handtracker.mpi-inf.mpg.de/projects/WatchSense/`
34. Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. 2015. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3213–3221.
35. Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. 2011. Fast articulated motion tracking using a sums of gaussians body model. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 951–958.
36. Emi Tamaki, Takashi Miyaki, and Jun Rekimoto. 2009. Brainy hand: an ear-worn hand gesture interaction device. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM, 4255–4260.
37. Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, and others. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 143.
38. Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. 2017. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 244.
39. Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-Meshes for Real-Time Hand Modeling and Tracking. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)* (2016).
40. Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)* 33, 5 (2014), 169.
41. Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016a. TouchRing. *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '16* (2016), 891–898. DOI: `http://dx.doi.org/10.1145/2957265.2961860`
42. Hsin-Ruey Tsai, Cheng-Yuan Wu, Lee-Ting Huang, and Yi-Ping Hung. 2016b. ThumbRing: private interactions using one-handed thumb motion input on finger segments. *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct* (2016), 791–798. DOI: `http://dx.doi.org/10.1145/2957265.2961859`

43. Hsin-ruey Tsai, Te-Yen Wu, Da-Yuan Huang, Min-Chieh Hsiu, Jui-Chun Hsiao, Yi-Ping Hung, Mike Y. Chen, and Bing-yu Chen. 2017. SegTouch: Enhancing Touch Input While Providing Touch Gestures on Screens Using Thumb-To-Index-Finger Gestures. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '17* (2017), 2164–2171. DOI: http://dx.doi.org/10.1145/3027063.3053109

44. Koji Tsukadaa and Michiaki Yasumurab. 2001. Ubi-finger: Gesture input device for mobile use. In *Ubicomp 2001 Informal Companion Proceedings*. 11.

45. Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. 2016. Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation. *International Journal of Computer Vision* 118, 2 (01 Jun 2016), 172–193. DOI: http://dx.doi.org/10.1007/s11263-016-0895-4

46. Dimitar Valkov, Ulrich Burgbacher, Klaus Hinrichs, and Computer Graphics. 2014. DigiTap: An Eyes-Free VR/AR Symbolic Input Device. (2014), 9–18.

47. Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2017. Crossing Nets: Combining GANs and VAEs With a Shared Latent Space for Hand Pose Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

48. Saiwen Wang, Jie Song, Jamie Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16* (2016), 851–860. DOI: http://dx.doi.org/10.1145/2984511.2984565

49. Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. Iskin: flexible, stretchable and visually customizable on-body touch sensors for mobile computing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2991–3000.

50. Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. 2017. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the 35rd Annual ACM Conference on Human Factors in Computing Systems*. ACM.

51. Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, and Ravi Karkar. 2017. DigiTouch : Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays. 1, 3 (2017), 1–21.

52. Christian Winkler, Julian Seifert, David Dobbelstein, and Enrico Rukzio. 2014. Pervasive Information Through Constant Personal Projection: The Ambient Mobile Pervasive Display (AMP-D). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 4117–4126. DOI: http://dx.doi.org/10.1145/2556288.2557365

53. Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. 2011. A taxonomy of microinteractions: Defining microgestures based on ergonomic and scenario-Dependent requirements. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6946 LNCS, PART 1 (2011), 559–575. DOI:http://dx.doi.org/10.1007/978-3-642-23774-4_45

54. Sang Ho Yoon, Ke Huo, Vinh P. Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-worn Textile Input Device with Multimodal Sensing in Mobile Interaction. *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14* (2015), 269–272. DOI: http://dx.doi.org/10.1145/2677199.2680560

55. Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2014. Plex. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (2014), 191–194. DOI: http://dx.doi.org/10.1145/2638728.2638746

56. Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

57. Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E Starner, Omer T Inan, and Gregory D Abowd. 2017. FingerSound: Recognizing unistroke thumb gestures using a ring. 1, 3 (2017), 1–19.