FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist

FANG HU, Cornell University and Shanghai Jiao Tong University
PENG HE, Cornell University and Hangzhou Dianzi University
SONGLIN XU, Cornell University and University of Science and Technology of China
YIN LI, University of Wisconsin-Madison
CHENG ZHANG, Cornell University

In this paper, we present *FingerTrak*, a minimal-obtrusive wristband that enables continuous 3D finger tracking and hand pose estimation with four miniature thermal cameras mounted closely on a form-fitting wristband. FingerTrak explores the feasibility of continuously reconstructing the entire hand postures (20 finger joints positions) without the needs of seeing all fingers. We demonstrate that our system is able to estimate the entire hand posture by observing only the outline of the hand, i.e., hand silhouettes from the wrist using low-resolution (32×24) thermal cameras. A customized deep neural network is developed to learn to "stitch" these multi-view images and estimate 20 joints positions in 3D space. Our user study with 11 participants shows that the system can achieve an average angular error of 6.46° when tested under the same background, and 8.06° when tested under a different background. FingerTrak also shows encouraging results with the re-mounting of the device and has the potential to reconstruct some of the complicated poses. We conclude this paper with further discussions of the opportunities and challenges of this technology.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; *Ubiquitous and mobile devices*; Mobile devices.

Additional Key Words and Phrases: hand reconstruction, pose recognition

ACM Reference Format:
Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 71 (June 2020), 24 pages. https://doi.org/10.1145/3397306

1 INTRODUCTION

Hand gesture recognition has been a well-established research topic in the human-computer interaction (HCI) community for years, as it enables a variety of interactive applications, such as input in VR or AR [19], accessibility [46], human robot interaction (HRI) [11], and wearable user interfaces [25, 55]. Many traditional hand pose estimation and gesture recognition technologies are based on computer vision using cameras placed in the

Authors' addresses: Fang Hu, fh292@cornell.edu, Cornell University, 239 Gates Hall, Ithaca, New York, 14853, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai, 200240; Peng He, ph369@cornell.edu, Cornell University, 239 Gates Hall, Ithaca, New York, 14853, Hangzhou Dianzi University, 1158 Baiyang Ave, Hangzhou, Zhejiang, 310018; Songlin Xu, sx237@cornell.edu, Cornell University, 239 Gates Hall, Ithaca, New York, 14853, University of Science and Technology of China, 96 Jinzhai Rd, Hefei, Anhui, 230026; Yin Li, yin.li@wisc.edu, University of Wisconsin-Madison, 6730 Medical Science Center, Madison, Wisconsin, 53706; Cheng Zhang, chengzhang@cornell.edu, Cornell University, 244 Gates Hall, Ithaca, New York, 14853.

2474-9567/2020/6-ART71 $15.00
https://doi.org/10.1145/3397306

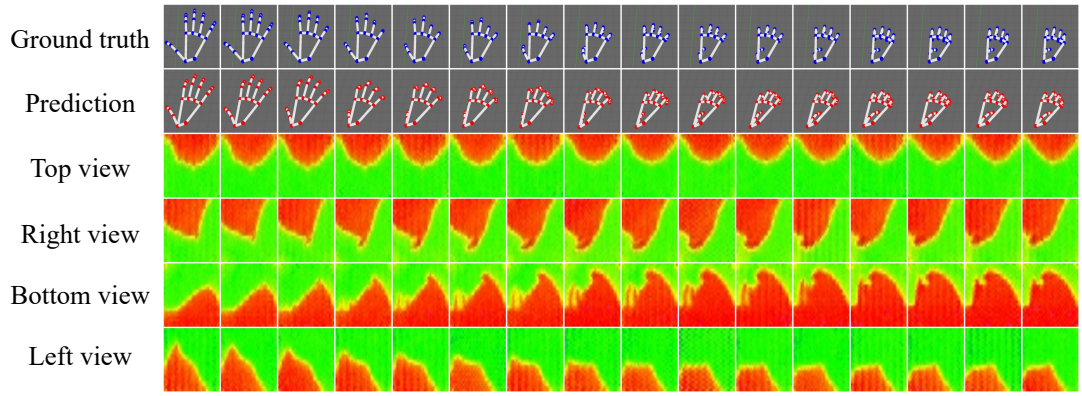

Fig. 1. Continuous hand pose estimation results using FingerTrak. Each column presents a result for a time step. From top to bottom: ground truth hand pose; predicted hand pose; and the thermal images captured by the wearable cameras.

background [4, 34, 35, 41, 49, 50], with the design that these cameras can capture the various views of the hand. Unfortunately, these technologies do not work well when the cameras cannot fully view the user's hand, and they also require a pre-instrumentation of the environment, which brings inconvenience for a user when facing a flexible situation. With the development of portable computers, especially wearables, there is a significant enrichment for the interaction scenarios, and the users need to interact with their computing interfaces using a more robust, convenient, and minimal-obtrusive device.

To address this need, many researchers developed wearable technologies with different sensing capacities to recognize hand postures. These projects can be categorized into two parts based on their recognition results: 1) discrete hand pose classification [53, 54, 56, 57]; and 2) continues hand posture reconstructions [25]. The majority of the prior work on wearable gesture recognition falls into the first category, and seeks to distinguish a pre-selected set of discrete hand poses or gestures using different sensing modality such as acoustic [54], cameras [25, 53], bio-impedance [56], pressure sensor [33]. However, these technologies cannot reconstruct the entire posture of the fingers. As a consequence, the applications of the these technologies largely depend on the pre-selected gestures that are challenging to be generalized to others.

Continuously reconstructing the hand posture provides a large potential for a number of applications. Unfortunately, most of these technologies that can reconstruct the complete hand postures use cameras in the background and does not work in mobile settings. Compared to discrete hand postures, it is very challenging to continuously reconstruct the entire hand postures using a wearable device, as it is hard to capture enough information about the hand. The most popular wearable technologies for full hand posture estimation make use of glove-like devices popular in professional settings such as movie industry. However, these glove-like devices are less practical and inconvenient for daily usage. Based on our knowledge, we have only observed a limited amount of prior work that can continuously reconstruct the entire hand postures using a glove-less wearable device. The only one that falls into this category is Digits [25]. Digits places an IR emitter and a camera on the bottom of the wrist to capture the fingers and extracts the finger positions from the captured images, based on which other finger joint positions can be calculated using inverse kinematics (IK). Digits requires the camera to see all fingers, and thus part of the camera system has to sit high enough on the wrist and even go beyond the wrist towards the palm. Furthermore, it also will not work when the fingers are blocked from the view of the camera when the hand is moving or rotating or holding objects.

The major challenge of all the previous camera-based finger reconstruction system is that the cameras have to capture the entire fingers to estimate the entire hand posture. In order to help wearable cameras on the wrist to observe the complete set of fingers, the cameras usually have to sit on a relatively high position on the wrist, making the wearable system bulky and less practical/comfortable. Furthermore, if the hand is rotating or moving, the system may fail as the fingers may be blocked by the palm. This limits the expressiveness of the hand postures the system can recognize.

Our key research question is that *whether we can reconstruct the entire hand posture using a wearable camera technology, which does not require seeing all the fingers.* In order to explore this research question, we developed *FingerTrak*, a minimal-obtrusive wristband that enables continuous 3D finger tracking without the need to observe all fingers. FingerTrak consists of four miniature thermal cameras (9.3 mm x 9.3 mm x 5.7 mm) mounted closely on a form-fitting wristband (2 mm on top of the skin) which can provide the user with minimal distraction and is flexible for different backgrounds. Instead of directly capturing the position of fingers as all prior work did, FingerTrak takes the advantage of the four cameras on the wrist to capture the outline of the hand, which we find very informative on estimating the entire hand posture including 20 finger joints positions. The captured images are sent to a customized deep neural network that learns to "stitch" the captured images from multiple views and estimate 20 finger joints positions in 3D space at a frame rate of 16 Hz. A user study with 11 participants showed that FingerTrak can continuously reconstruct the 20 finger joint positions with an average displacement error of 1.2 cm. We also evaluated FingerTrak under a variety of settings, including different backgrounds, different arm postures and when the user's hand is holding objects. To our knowledge, we are the *first wearable technology* that attempts to reconstruct the entire hand posture when the hand is holding objects. Furthermore, we discuss the limitations and potential improvements of our technology in the paper.

The contributions of the paper are:

- We developed a wearable system, using a wristband with four miniature thermal cameras (1.19 cm above the skin) to capture the outline of the hand. We demonstrate that we can estimate the 20 finger joints positions (entire hand) in 3D space by only observing the outline of the hand (hand silhouettes) from the wrist using a deep neural networks.
- We conducted a user study with 11 participants to evaluate the performance of the system under different scenarios, including different backgrounds, different arm postures and holding objects in the hand.
- We discussed the opportunities, challenges and limitations of applying FingerTrak in real-world applications.

In the rest part of the paper, we will review previous work and highlight the innovation of FingerTrak. We then present the underlying theory, design and implementation, and empirical evaluation of the system. Finally, we discuss the opportunities and limitations of this novel technology.

2 RELATED WORK

Hand pose estimation has been a focus among communities of human-computer interaction, computer vision and graphics. Researchers have explored various sensing modalities and form factors, and the placement of sensors to address this research challenge. We discuss methods that estimate hand poses using external sensors and wearable sensors, followed by a review of the most relevant work on sensing hands using wrist-mounted devices.

2.1 Hand Posture Recognition Using Non-wearable Sensing

Camera-based hand posture estimation has received considerable attention in both computer vision and graphics communities. Commercial marker-based motion capture systems, such as Vicon,[1] have been developed for hand tracking. These systems require not only the placement of markers on the body of the user but also heavy instrumentation of the environment. Recent efforts are thus focused on marker-less hand tracking. Given a

[1] https://www.vicon.com/

prior 3D model of the hand, a user's hand pose can be reconstructed by fitting the model with images from either multiple cameras [35] or a single depth camera [34, 37], or even a monocular camera [9]. More recently, researchers started to apply machine learning models to directly estimate hand postures from images [14, 22, 41–43, 49, 50, 59]. Learning-based methods incorporating the shape and pose of hands has demonstrated improved results to approaches with no prior information about the hand. Nonetheless, these vision-based methods require externally fixed cameras and, as a result, limiting these methods for daily interaction and making these technologies less suited to mobile and ubiquitous uses.

Going beyond cameras, Li et al. [29] have presented a system for tracking hand postures using an imaging system that combines a LED array in a lampshade and multiple photodiodes on the base of the lamp. However, their sensing range is limited to the size of a lamp. To enable hand tracking over a more extensive range, researchers explored several new methods using Wi-Fi [23, 48] or acoustic sensors [36] for hand sensing. However, these methods can only recover the motion of hands but not the posture of fingers.

2.2 Recognizing Hand Postures from Non-wrist Mounted Wearable Devices

Mounting sensors on the user's body removes the need for external sensors, thus allowing applications to interact in mobile settings. The most common form factor that has been used for hand pose tracking is gloves. These gloves usually are embedded with multiple sensors to capture the motion of the palm and fingers, and these signals are further assembled into full 3D hand pose by using optimization techniques. For example, gloves based on inertia measurement units (IMU) have been explored in [8, 31]. Other sensing modalities have also been considered, including bend (flex) sensors gloves [7, 27], strain sensors gloves [6, 15] and stretchable sensor arrays [16]. Glove providing a good solution for many application settings, such as motion capturing in movie industry. However, many people may still not be comfortable wearing gloves in daily activities.

Other than gloves, Rogez et al. showed that a chest-mounted depth camera can be used to estimate the user's hand pose [38]. However, the chest-mounted camera can only capture the hands when they are in front of the chest and fully exposed to the camera. Discrete hand gestures can also be recognized by using a shoe-mounted camera [1] and a handheld device [40]. Another appealing approach for hand tracking [30] is to use head-mounted cameras widely available in many off-the-shelf commercial devices such as Oculus Quest, HTC Vive Pro, and Microsoft Hololens.[2] Nonetheless, these head-mounted devices are designed for AR/VR applications and cannot be used to estimate hand pose without the headset, making them less convenient for every day use.

2.3 Recognizing Hand Postures Using Wrist-mounted Devices

In comparison to other form factors, users tend to be more acceptable to wrist-mounted devices such as smartwatches and wristbands, as people are used to wear watches for years. Therefore, many wrist-mounted devices have been developed to recognize hand gestures. For example, the sensors in a commodity smartwatches was explored for recognizing discrete hand gesture [52, 58]. Customized wrist-mounted devices with different sensing modalities were also built for hand pose recognition, including wrist-worn pressure sensors [10], infrared proximity sensors [26], distance sensors [12], surface electromyography sensors [24], and active acoustic sensors [33, 54]. However, these techniques can only recognize discrete gestures, which is limited in many scenarios.

Cameras can be also mounted on the user's arm [44] or wrist [51, 53] to recognize two-handed or one-handed discrete poses and gestures. The most relevant work is Digits [25]. Digits mounted an active IR camera system on the bottom of the wrist to capture the positions of fingertips, and used inverse kinematics (IK) to estimate the positions of other finger joints. However, Digits requires the IR camera on the wrist to always seeing the positions of all fingers. Therefore, the camera has to sit high enough on the wrist and even go beyond the wrist towards the palm. Apparently, it will not work when the fingers are blocked from the view of the camera. For

[2]https://www.oculus.com/, https://www.vive.com/us/product/, https://www.microsoft.com/en-us/hololens/

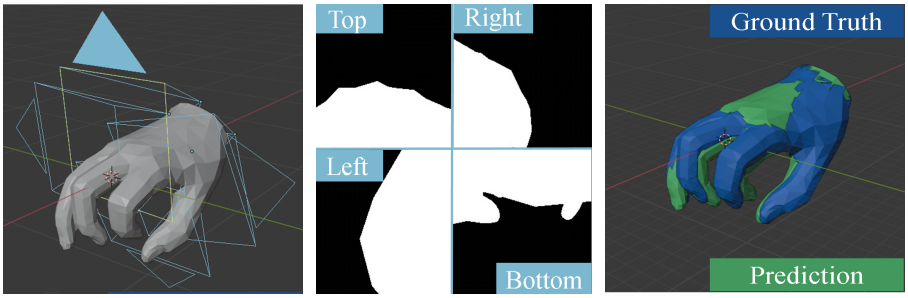

Fig. 2. Sample images and results from our synthetic dataset. Left: an example of a synthetic 3D hand at a random pose and shape; Middle: hand silhouettes captured by 4 virtual cameras; Right: A comparison between the predicted 3D hand pose (green) and the ground truth (blue).

instance, when the hand moves upward from the wrist or holds objects in the hands. This significantly limits the expressiveness of the hand postures the system can recognize. The latest work that shares a similar spirit with FingerTrak is from [53]. Their method observes the outline on the back of hand using a commercial depth camera (Leap Motion). Their camera is sitting at a much higher position on the wrist, and can only recognize discrete gestures. In comparison to [53], FingerTrak is not only smaller, more comfortable, but also more capable.

FingerTrak is the first wrist-mounted device that can reconstruct the entire hand pose (20 finger joints positions) by learning the outline shape of the hand captured by 4 miniature thermal cameras sitting tightly on the wrist. A key innovation of FingerTrak is that it does not require the system to observe the fingers to reconstruct the entire hand pose. It is thus possible to estimate the hand poses even when the hand is occupied with objects, as we will demonstrate in our user study.

3 FINGERTRAK: FROM THEORY TO PRACTICE

Consider a set of hand silhouettes (outlines of the hand) captured at different viewpoints, e.g., those in Figure 2 (middle). Can we estimate the hand pose from those silhouette images? The problem is fundamentally ambiguous as different hand poses might produce the same silhouette due to viewpoint variation and self-occlusion. However, the ambiguity can be largely reduced by fitting a prior 3D hand model to multiple silhouettes from different angles, where the 3D model and multi-view hand contours constrain the underlying 3D shape of the hand.

This idea of using silhouettes for 3D shape reconstruction, also known as Shape-From-Silhouette (SFS), was first discussed by Baumgart [5] in 1974. SFS has since received considerable attention in computer vision and graphics communities [2], and has been used for markerless human body motion tracking [3]. We build on the idea of using SFS for markerless human tracking, and develop a wrist-worn wearable system for continuous 3D hand posture tracking. At the core of our system lies in the idea of using multiple wrist-worn cameras to capture hand silhouette images, and to further reconstruct the 3D pose of the hand. Our research hypothesis is that *it is possible to accurately estimate 3D hand posture by only combining multi-view hand silhouettes(outline) with the prior knowledge of a 3D hand model.*

To evaluate the feasibility of our hypothesis, we start with a synthetically rendered image dataset, which was generated by virtual cameras on the wrist of a 3D hand mesh model [39], as shown in Figure 2. Furthermore, we develop a deep neural network to estimate hand pose from these synthetic images. The initial result was very encouraging, which further motivated us to design and implement the prototype of FingerTrak. In the reset part

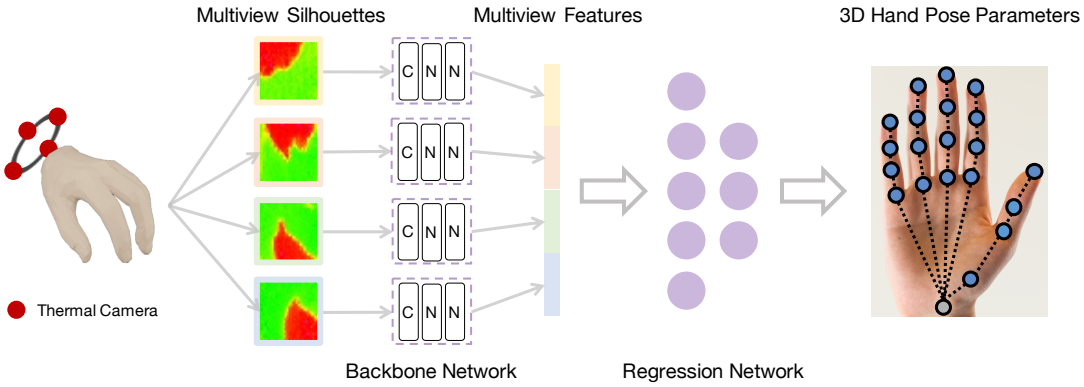

Fig. 3. Our deep model for 3D hand pose estimation. The model takes multiple hand silhouette images captured by wrist-worn cameras as input. Each image is feed into a convolutional neural network (CNN). Their features are concatenated and sent to a regression network, which outputs the parameters of 3D hand pose.

of this section, we will present the details on the creation of synthetic hands, our deep learning model, and the design and implementation of the FingerTrak system.

3.1 Generating Synthetic Multi-view Hand Silhouettes

As a starting point, we create a pipeline for generating synthetic hand images to facilitate the verification of our research hypothesis. To this end, we make use of MANO—a parametric model for hands [39]. MANO contains two sets of parameters that control hand shape and pose respectively. In order to simulate different hand shapes and postures, we randomly sample these shapes and pose parameters from a Gaussian distribution, place different number of virtual cameras on the wrist (equally spaced) and render the resulting multi-view hand images using Blender.[3] As a proxy to hand silhouette images, we consider a black background with a white foreground hand. A major advantage of using synthetic hand images is that we can easily control number of cameras and the position of the cameras to generate millions of pictures to verify critical system design decisions, including the number and positioning of cameras, the number of training samples and the neural network design. Figure 2 (middle) shows an example from our synthetic data set captured by 4 virtual cameras.

3.2 From Multi-view Hand Silhouettes to 3D Hand Pose Using Deep Neural Networks

Going forward, we make use of multi-view convolution neural networks (Multi-view CNN) [47] for 3D hand pose estimation. Multi-view CNN was originally designed for 3D shape classification and retrieval. We re-purpose the model to regress the parameters of 3D hand pose. Concretely, our model learns to stitch the multiple frames K and predict a $\mathcal{J}$-dimensional continuous output that defines the 3D hand pose. For real world data, we present the hand pose using the 3D coordinates of all 21 hand joints and locate the wrist position as the original point, thus $\mathcal{J} = 3 \times 20$ is needed for the rest 20 finger joints. For synthetic data, we use a equivalent parametric model with $\mathcal{J} = 9$ from [39]. The model includes a backbone network and a regression network. Each of the K frames is sent to the backbone network, where their features are extracted independently. These features are further concatenated and fed into a regression network that predicts the hand pose parameters. An illustration of our deep model is shown in Figure 3.

[3]https://www.blender.org/

Network Architecture. Our backbone network follows the same design of the convolutional blocks in a 34-layer residual network [17] (ResNet-34), which has been proven highly effective for visual recognition tasks and less prone to over-fitting. A convolutional block in ResNet includes several convolution operations, each followed by batch normalization [21] and rectified linear unit (ReLU). A global average pooling is performed at the end of the backbone to extract a vector representation of each image. Features from each view is further concatenated and sent to a regression network. Our regression network consists of two fully connected layers with ReLU in-between and a dropout [45] ($p = 0.5$) before the last layer. The regression network thus maps the multi-view features into a continuous output of hand parameters.
Model Training. Our model was trained with ground truth hand poses using Huber loss [20] (robust regression). We used standard mini-batch stochastic gradient descent (SGD) with momentum (0.9), weight decay (1e-4), batch size 256 and a learning rate of 0.001. Cosine learning rate annealing [32] was also used. The choice of these hyper-parameters were chosen based on the common practice established in the vision community [18]. We did not employ data augmentations during training, such as flipping or cropping, as they will not preserve the 3D geometry of the hand. For all experiments, our model was trained for 90 epochs on the training set, with each epoch is a pass over the full training set. The trained model was further evaluated on a hold-out non-overlapping test set.

3.3 Towards Hardware Prototype Design: A Study of Hand Pose Estimation Using Synthetic Data

We now present a study of 3D hand pose estimation from multi-view hand silhouettes by using the deep model and the synthetic data generation pipeline. Our goal is to identify a good design for our hardware prototype. To this end, we vary the number of cameras and their positions, render the synthetic hand images, train deep models and evaluate their results.
Camera Settings. We considered 8 equally spaced slots around the wrist (see Fig 4) and experimented to position different number ($K = [1, 2, ..., 8]$) of virtual cameras[4] into these slots. A brute force search will need to cover $2^K = 256$ combinations. We instead designed a greedy search strategy to identify the best camera mountings. Specifically, we started with a single camera and enumerate all of the 8 options. For each position, we trained a model using the rendered data and select the position with the lowest model prediction error on the test set. In the next round, another camera was added to one of the rest slots, and again the model prediction error was used to choose the best position. This process was repeated until all slots are filled.

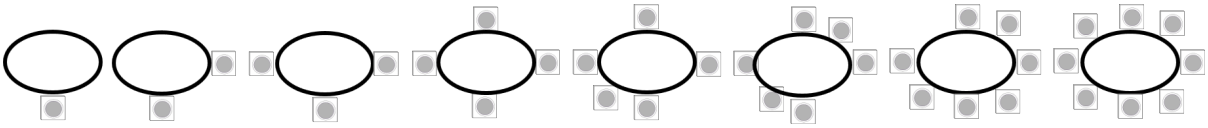

Fig. 4. Best configuration of camera position given by our greedy search. From left to right, camera number varies from 1 to 8. Right hand is used for this visualization and we assume that the hand is pointing to the paper with the palm facing downwards. Thus, a bottom camera is on the palm side while a top camera is on the back of the palm. Similarly, the left camera is on the thumb side while a right camera is on the little finger side.

Synthetic Data Generation. For each camera setting, we rendered a large-scale dataset divided into a training set of 600, 000 images and a non-overlapping test set of 10, 000 images. Both training and testing set were randomly sampled and cover a wide range of hand poses and shapes. A small Gaussian noise was added to the position and the orientation of each camera to simulate the slip of the sensors on the wrist.

[4]Each camera has a field of view (FoV) of 110°, matching our thermal camera.

Table 1. Reconstruction error of the best camera configurations as shown in Fig. 4.

Camera number	1	2	3	4	5	6	7	8
MAE (cm)	1.67	1.01	0.72	0.65	0.63	0.58	0.55	0.53

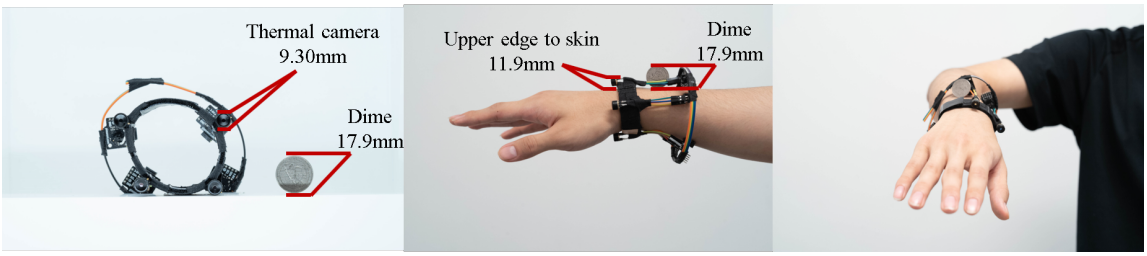

Fig. 5. An illustration of our hardware prototype and its mounting. Left: The prototype wristband sitting on a table and compared to a coin. Middle: The mounting of the same prototype on the wrist and compared to the same coin. Right: A different view of the mounting. We annotate the size of the camera and the coin by millimeter (mm).

Hand Pose Results. Given K cameras, our deep model takes K rendered hand images from each camera, and seeks to regress the MANO pose parameters ($\mathcal{J} = 9$) that are used to create the 3D hand. In this setting, we evaluate our model using mean absolute error (MAE) over the 20 joints. The MAE is converted into a physical metric space (cm) by re-scale all data points to the mean shape of hands. The result configurations from the greedy search are shown in Figure 4, where the best camera mounting is displayed for each $K = 1, 2, ..., 8$. Moreover, Table 1 presents the MAE for all configurations in Figure 4. Out of the 8 options, using a bottom camera on the palm side leads to the best MSE while using a top camera on the back of the palm has the worst MSE. We speculate that the hand silhouettes from the bottom view captures a significant portion of the finger movement.

Surprisingly, even with a single camera facing the palm, the deep model achieves an MAE of 1.67 cm. The reconstruction error indeed keeps decreasing as we add more cameras. For example, with 4 cameras the MAE is 0.65 cm. A sample result using 4 cameras is shown in Figure 2 (right). We also observed a diminished return of the reconstruction performance when adding more cameras—adding another 4 cameras (8 in total) only reduces the MAE to 0.53 cm. These results provide important guidelines for our hardware design. With our goal of accurate reconstruction using a minimum set of cameras, we choose to use 4 cameras for implementing our hardware prototype.

3.4 FingerTrak: System and Implementation

The pose estimation result from the synthetic data set was very encouraging. Therefore, we move forward to design and implement our physical prototype FingerTrak. We now present the details.

3.4.1 Hardware Design. Our prototype consists of a wristband module, four thermal cameras placed at the designed location, and a computing unit, as shown in Figure 5.

Design of the Wristband. The wristband is made of Velcro, such that the size of the band can be adjusted to fit different sizes of wrists. Each Velcro has a length of 17cm and a width of 3cm. We taped four thermal cameras on the wrist bands with equal distances in between. We chose thermal cameras because they allow the system to reliability capture hand silhouettes even under complex backgrounds. We considered and experimented different cameras in our preliminary study. A depth (RGBD) camera can potentially easily segment hands from the background, but its size is relatively large at this moment, which does not meet our design requirements.

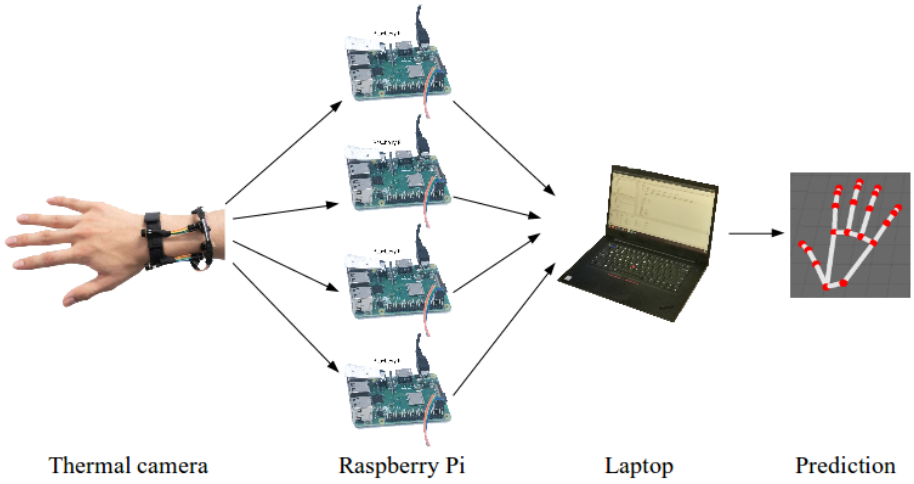

Fig. 6. Overview of the system architecture. Four Raspberry Pi's transfer synchronized thermal images to the laptop via a Wi-Fi router. The laptop retrieves a stream of frames containing tracking data and predicts the hand pose.

Compared to thermal cameras used in our system, miniature RGB cameras can provide higher resolutions. However, reliably hand segmentation under cluttered background poses extra challenge in computer vision. As the research question in this paper is to explore whether we can reconstruct the complete hand pose from the outline of the hand, we decided to use the thermal cameras in our final prototype. We will discuss more detailed results with other camera choices in the discussion section.

Implementation Details. Four MLX90640 thermal cameras were used as the main sensors in our prototype. Each camera comes with a thermal sensor (CCD and lens) and a image processing unit (integrated in a small PCB), and is 9.30 mm in radius and 11.25 mm in height. We re-wired the connection between the thermal sensor and its processing unit, such that only the sensor was mounted on the wristband, further reducing the size and allowing for a mounting position that is closer to the wrist. These cameras can continuously record thermal images of the hand with a temperature sensitivity around $\pm 2^\circ$ C (in the $0 - 100^\circ$C range) at a frame rate of 16Hz with a resolution of 32×24, and a field of view (FoV) of $110^\circ \times 75^\circ$. Each camera was connected to a separate Raspberry Pi 3B+ through I2C interface. In total, 4 Raspberry Pi's were used to communicate the captured image frames from the camera (16 FPS) to a ThinkPad X1 laptop with Intel Hexa-Core i7-8750H CPU through wireless local area network. Figure 6 shows the full hardware setup of our system including the wrist band, Raspberry Pis and a laptop.

Image frames from our prototype system were further synchronized using a time-matching algorithm. Concretely, UTC time stamps from the Raspberry Pi were used to match the frames. Thermal images were collected by Raspberry Pi together with their time stamps into a buffer. A threshold (0.1 second) was selected such that four thermal images are considered as aligned if and only if the absolute maximum difference among their time stamps is lower than the threshold. We removed all frames that are not aligned based on our time-matching algorithm. Therefore, our prototype system might miss a few frames during the recording.

In addition to our hardware prototype, we mounted an external depth sensor—Leap Motion to capture hand poses during our user study as a way to train and evaluate our system. Specifically, Leap Motion utilizes raw depth images captured by its depth cameras to infer a reference hand pose at a frame rate of 60Hz. Its major

hardware components include three active infrared LED's and two depth cameras (Aptina MT9V024 with global shutter) receiving the reflection of the infrared light. A depth image records the distance of each detected object point from the camera's viewpoint. Estimation of reference pose (3D coordinates of hand joints) is based on a proprietary finger joint detection algorithm applied to depth images captured by the imaging sensors. Hand pose data is then transferred to a laptop via USB and received by a program calling Leap Motion API. The program uses Leap Motion V2 desktop developer SDK. It receives a stream of hand pose data, including hand joint positions in Leap Motion coordinate system and timestamps in UTC. After data collection, recorded hand pose data is synchronized with thermal images.

We further synchronized thermal images from our prototype system to the reference pose acquired by Leap Motion using a similar time-matching algorithm as we synchronized the thermal images. As Leap Motion also creates time stamps for its pose data and has a higher sampling rate than our prototype, we chose to match the time stamps from Leap Motion to their nearest time stamps from Raspberry Pi. Similarly, a threshold was also selected such that the thermal images are considered as matched to a reference pose if and only if the absolute maximum difference their time stamps is lower than the threshold. After this synchronization, a set of thermal image frames with reference pose data were saved. We considered these reference pose as the ground-truth pose for the corresponding thermal images.

To train our deep model for hand pose estimation, the thermal images and their ground-truth data were transmitted to a GPU cloud instance (p2.8xlarge instance with 8 K80 GPUs) at Amazon Web Service (AWS). Once trained, the model can be deployed to a workstation to conduct pose estimation using our prototype.

3.4.2 3D Hand Pose Estimation Using FingerTrak. We further integrate the deep model with our hardware for estimating 3D hand pose based on multi-view thermal images. Our system continuously captures hand images from the wrist. At each time step, our model takes the input of $K = 4$ thermal images (time synchronized) of the size 32×24, and outputs the 3D coordinates of 20 joints (output dimension $\mathcal{J} = 60$).

Hand Representation. We directly used the thermal images as the input. While it is possible to segment the hand regions based on body temperature, we found our model works reasonably well even without segmentation. For the output, we regress 60 values that define the 3D coordinates of 20 joints, shown as the blue dots in Fig 3 (right). We exclude the root joint of wrist. This is because our cameras always move along with the wrist and thus the wrist joint is considered as the origin of our coordinate system. This hand model was also used in [39]. We note that choice of predicting 3D coordinates of all joints (an over-parameterized representation of the hand) is deliberate, as these 3D coordinates naturally encodes the length of phalanx bones.

Training with FingerTrak Data. Our model was trained using synchronized ground-truth pose data gathered from Leap Motion, as we described previously. Specifically, our model takes 4 thermal images captured from our prototype and learns to predict hand poses (20-joint hand representation) measured by Leap Motion. We experimented with training both user dependent and user independent models as we will describe in our user study and discussions. All models were trained from a random initialization (scratch). We followed the same training scheme as presented in Section 3.2 yet with a different mini-batch sampling strategy. We found it helpful to avoid sampling adjacent frames in the same mini-batch during training, as these frames tend to be nearly identical and thus slow down the learning. Once trained, our model can be applied on every frame of an input video, leading to continuous estimation of 3D hand pose.

4 USER STUDY

To better understand the performance of FingerTrak with users, we conducted a user study with 11 participants (6 males and 5 females) to evaluate our system. All participants were recruited from the local institution, and the study was approved by Institutional Review Board (IRB).

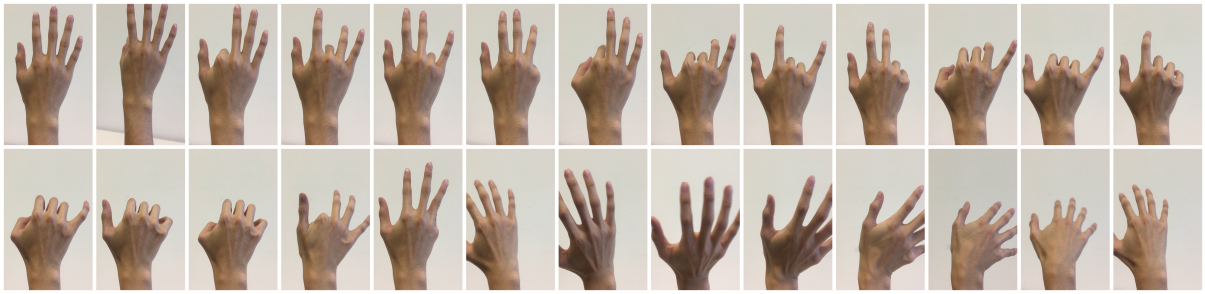

Fig. 7. Examples of the continuous hand poses used in the first section of our user study.

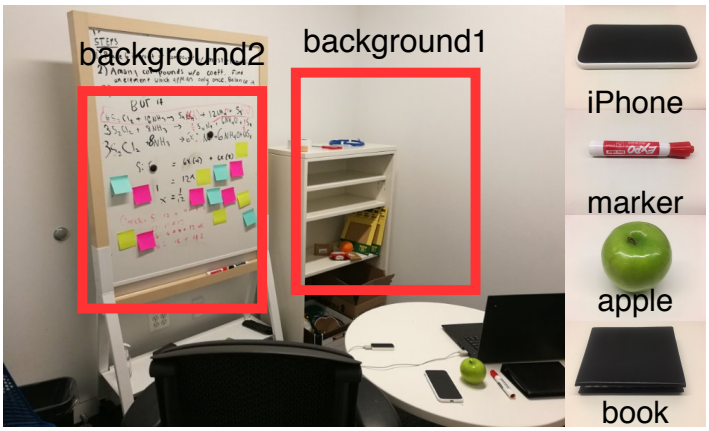

Fig. 8. Two different backgrounds (left) and four items used for holding during our user study (right).

4.1 Study Procedure

At the beginning of the study, a researcher introduced the protocol of the study to the participants and answered any question they may have about the study. Then one researcher helped the participant to wear the wristband. Participants were asked to choose which hand (left vs. right) they want to wear the wristband. All chose right hand. Furthermore, the participants were encouraged to practice moving their hands/fingers with the wristband to ensure comfortableness, before the study starts.

During the study, each participant was sitting on a chair. The Leap Motion was placed on the table. To acquire reliable ground truth hand postures, we expect that the participants' hand stay in the field of view of Leap Motion. In order to help the participants hold the hand within the zone, we put a comfy box on the table, where the participants can put their elbow on. It helped to keep the arm above the table while reducing the fatigue. During the study, the participants can move their arms and hands, as long as the hand is in the range where the Leap Motion can capture the hand postures.

The study for each participant consists of two sections, which are designed to evaluate the performance of hand pose estimation under two different settings: when the hand is empty or when the hand is holding objects. The entire study for each participant lasts for around 1 hour. We now present details of each section.

4.1.1 Continuous Hand Pose Estimation When the Hand Is Empty. In the first section,the participant performed different hand postures following the pre-recorded instructional video which was played using a monitor on the table. In this instructional video, the researchers demonstrated performing the 19 hand poses (as shown in Figure 7) slowly in a random sequence. Before and after performing each pose, the hand always returned to the original pose ,which is the left-most pose on the first row in Figure 7. The guideline for choosing the postures used in the instructional video is that we want to choose the postures as complicated as possible. However, we found that Leap Motion cannot provide reliable ground-truth for complicated postures. After balancing these factors, we chose the final 19 postures which can be recognized by Leap Motion as shown in Figure 7. We randomized the sequence of 19 postures and generated 3 instructional videos.

This section had 10 sessions in total. In each session, the participants performed the gestures following one of the three videos. The first 7 sessions were used as the training sessions. The rest of the sessions were used as the testing sessions. To evaluate how would our system work under different background, we asked the participants to rotate the arm orientation to a different direction on the table with a different background in the second to the last session (see Figure 8 for the background). In the last session (10th), the device was first removed from the wrist of participant and re-mounted again. In each training session we adjust the wristband to a slightly different position and orientation, so that the data sampled can bear more variance and make the trained neural network more generalizable. The wristband is put back on with our assistance to make the position shift as small as possible. In the future, calibration gestures may be involved after re-mounting, which will be further discussed in Section 5.5. The background was kept the same as the first 7 sessions.

4.1.2 Hand Pose Estimation When the Hand Is Holding Objects. The goal of the second section of the study is to evaluate how the system performs on estimating hand poses when the hand is holding objects. Prior work using wrist-mounted cameras [25, 53] for hand pose estimation all required the complete view of the fingers. Thus, they cannot estimate the hand pose when the hand is holding objects, as the object will block the fingers from the wrist-mounted camera. Based on our knowledge, our work provides the first attempt to estimate the complete hand pose using a wrist-mounted camera when the hand is occupied with objects.

We chose four common objects (a smartphone, a book, an apple and a sharpie as shown in Figure 8) for the participants to grab and hold in this task. This section has 40 sessions. In each session, the participants grabbed and held each of the four objects for about 4 seconds (50 frames) in a random order. The first 32 sessions were used as training sessions and the last 8 sessions for testing session. Leap Motion cannot provide reliable hand posture estimation as ground truth when the hand is holding objects. Therefore, after the participant grabbed and held the object, we removed the object from their hands, and asked the participants to keep the same posture as stable as possible, such that Leap Motion can provide ground-truth of the hand poses.

In this section, we only estimated the static hand pose when the hand was holding the object due to the limitation of ground truth acquisition. The training data are the images captured from the four wrist cameras when the user is holding objects in a static pose. And the ground truth hand pose is provided by Leap Motion, when the object is removed from the hand while keeping the same static pose.

4.2 Results of Continuous Hand Pose Reconstruction

The data gathered from our user study were used to train and evaluate our model. For each participant, we obtained an average of 10K samples for training and 2K samples for testing, leading to a total of $(10K + 2K) * 11 = 132K$ samples. Each sample contains 4 thermal images captured at the same time. These 132K samples were sampled key frames from the videos during the sessions of the user study and is equivalent to 750 seconds of videos with a 16Hz frame rate. For our experiments, we trained a separate model per user, i.e., our models are user-dependent.[5]

[5]See the experiment of user independent models in Section 5.7.

Fig. 9. Results of our user study. We report per joint MAE for joint positions (left column in cm) and angles (right column in degree). From top to bottom: testing under the same background (first row); testing under a different background (second row); testing after re-mounting the sensors (third row); and testing with an object in hand (last row).

These models were evaluated for continuous hand pose estimation. And the results were reported using both position and angular errors. We now describe our evaluation protocol and present our results.

Evaluation Protocol. We consider two different settings in our user study: (1) when the hand is empty and (2) when the hand is holding objects. For both settings, we compare our model outputs to reference hand poses obtained from Leap Motion. The mean absolute error (MAE) for joint positions and joint angles are calculated as

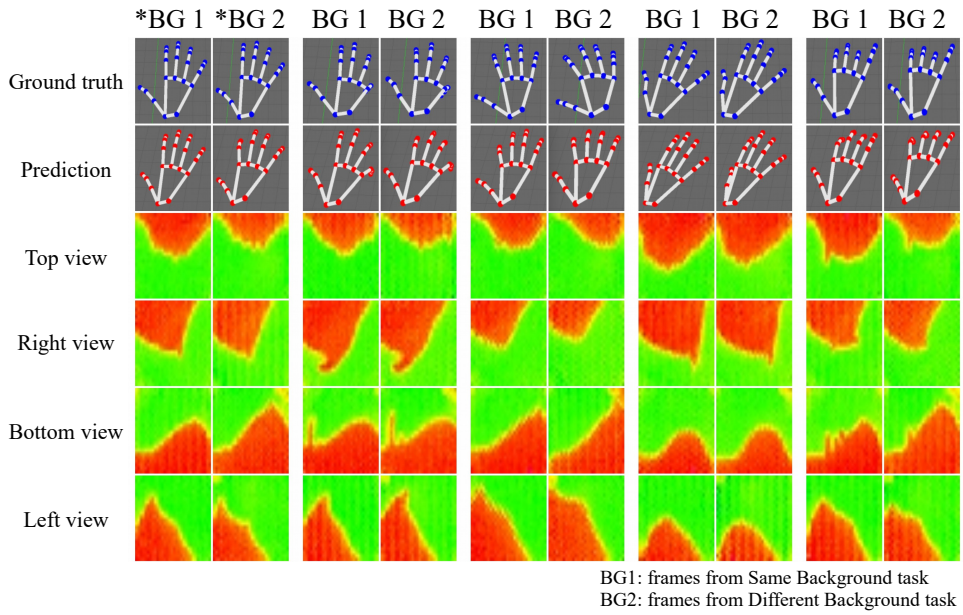

Fig. 10. Visualization of the results with different backgrounds (BG1 vs. BG2). Each column presents two sample inputs with a similar hand pose and compares the results from thw two backgrounds (BG1 left and BG2 right). From top to bottom in each column: From top to bottom: ground truth hand pose; predicted hand pose; and the thermal images captured by the wearable cameras.

our main evaluation metric. Specifically, for each of the 20 joints (see Figure 3 right), the absolute error between the predict 3D coordinates and the reference 3D coordinates are computed. To further capture the 3D structure of the hand, we also compute the error between the predicted 3D joint angle and the reference joint angle on 15 joints (excluding the fingertips). The errors are further averaged across all joints and all frames, leading to MAE for joint positions (cm) and joint angles (degree).

4.2.1 Hand Pose Estimation When the Hand Is Empty. For the setting of empty hands, we use the data from the first 7 sessions for training. These sessions were captured under the same background. Our trained model was evaluated on a separate set of 3 sessions. Each of them is designed to evaluate a different condition. We present and discuss the results for each setting.

Same Background. As the first step, we evaluated our model on test images that were captured under the same background. Our results are presented in Figure 9 (first row). Overall, our system achieves an MAE of 1.20 cm for joint positions and a MAE of 6.46° for joint angles across all 11 participants. For reference, one of the latest computer vision methods (user-independent model) [13] for hand pose estimation using a depth image has a mean error around 1 cm for joint positions. And a recent wearable stretch-sensing glove can achieve a mean error about 6° for joint angles using a user-dependent model [15]. Moreover, our system can consistently predict accurate hand pose across participants (std MAE is ±0.31cm and ±2.12° for positions and angles).

To better understand our results, we plot the cumulative distribution function (CDF) of errors in Figure 12 (session 1). The curves demonstrate the percentage of samples where its worst joint prediction error (out of 20 joints) is smaller than a threshold. More than 50% of the samples have a worst error smaller than 2.5 cm—a small to moderate error for hand pose estimation. And all predicted joint positions in over 81% of the frames have no

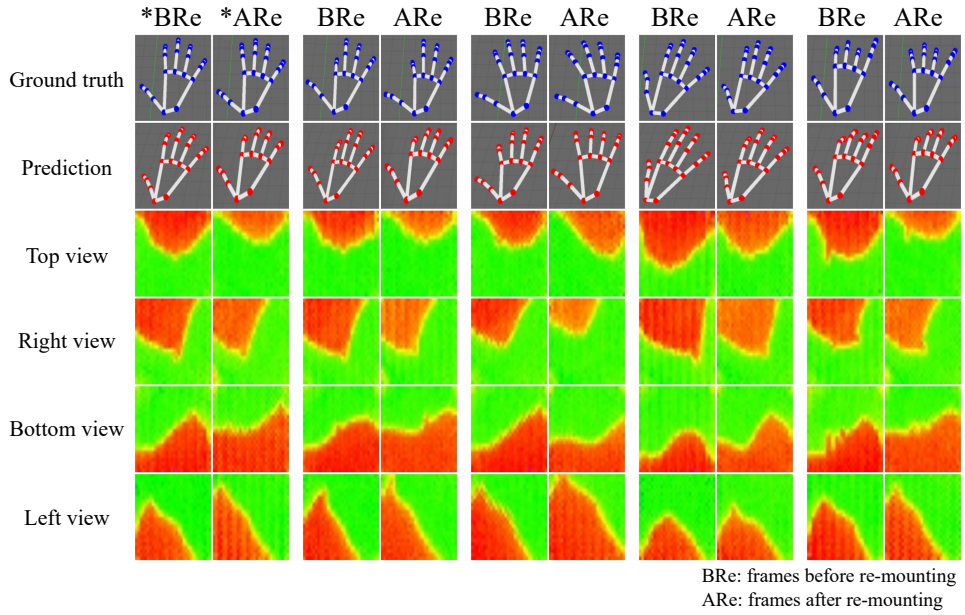

Fig. 11. Visualization of the results before the re-mounting (BRe) and after the re-mounting (ARe). Each column presents two sample inputs with a similar hand pose and compares the results before (left) and after the re-mounting. From top to bottom in each column: From top to bottom: ground truth hand pose; predicted hand pose; and the thermal images captured by the wearable cameras.

worse than 4 cm error. Finally, we provide visualization of our sample results in Figure 1, including the reference pose, predicted pose, and the input thermal images.

Different Background. Going forward, we further tested the same model on test images captured under a different background. Figure 9 (second row) presents our results. In this more challenging setting, our system has MAEs of 2.09cm (std ±0.58cm) and of 8.06° (std ±2.83°) for joint positions and angles, respectively. Switching to a different background increases the error by 0.89 cm and 1.6° in position and angle. Similarly, we plot the CDF of errors in in Figure 12 (session 2). In this more challenging setting, more than 50% of the samples have a worst error smaller than 5 cm.

We empirically observed that the captured thermal images are robust to different backgrounds. And we conjecture that the increased error was due to the shift of the cameras during large arm motion. This is further demonstrated by the visualization of sample results in Figure 10. With similar hand poses, there seems to be a minor shift in the thermal images, especially in the bottom view. Nonetheless, our results demonstrate that our system is able to generalize to different background.

Re-mounting. Finally, we consider a more practical setting, where the device was taken off and re-mounted on the same participant. Re-mounting the device can cause potential shifts of the cameras. Therefore, this cross-session setting is very challenging for our user-dependent model. We tested the same model and summarize the results in Fig 9 (third row). Even in this setting, our system still maintains reasonable MAEs of 2.72cm (std ±0.38cm) and 9.44° (std ±3.32°) for joint positions and angles. Similarly, we plot the CDF of errors in Figure 12 (session 3) and provide visualization of sample results in Figure 11. These additional results indeed support our argument on camera shifts. Figure 11 shows minor to moderate difference of the input thermal images of similar

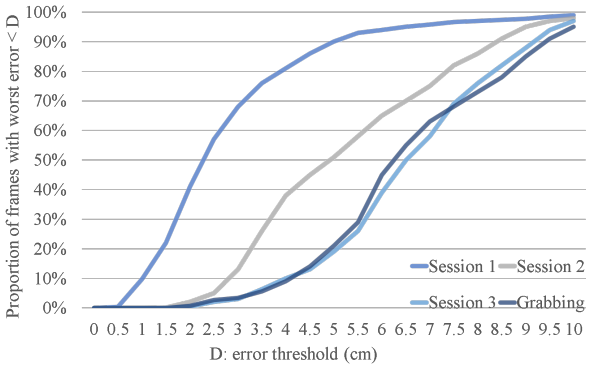

Fig. 12. The cumulative distribution function of the worst errors across all four settings. Session 1: same background; Session 2: different background; Session 3: re-mounting; and Grabbing: objects in hand. The X axis denotes the error threshold (D) in cm. And the Y axis show the percentage of frames where their worse joint prediction error is smaller than the threshold (D).

poses before and after the re-mounting. These differences, as we argued, lead to decreased MAE, where more than 50% of the samples have a worst error smaller than 6.5 cm. These results indicate that our system can adapt to minor variations in the mounting of the wristband, and thus have great potential for real world use cases.

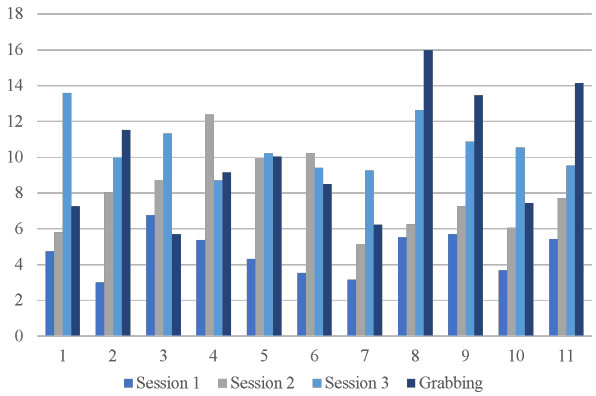

Fig. 13. MAE for joint angles for each participant. The average angular MAE across all three setting is 8.99±2.7 degrees.

Remark. We further discuss our results by contrasting three different settings. Testing our model under the same background gives the lowest errors that are comparable to other sensing modalities. Changing the background or re-mounting the sensor leads to a moderate increase in the pose estimation errors. We examined the source of the increased errors and attribute them to the shifts of cameras, which pose an additional challenge of inconsistent viewpoint for our user-dependent models. Figure 13 shows a breakdown of the angular MAE across all participants averaged across three settings. Among all settings, the thumb has the lowest error while the index finger has the highest. Overall, our system demonstrated promising accuracy for 3D hand pose estimation across participants, under different backgrounds and with the re-mounting of the device.

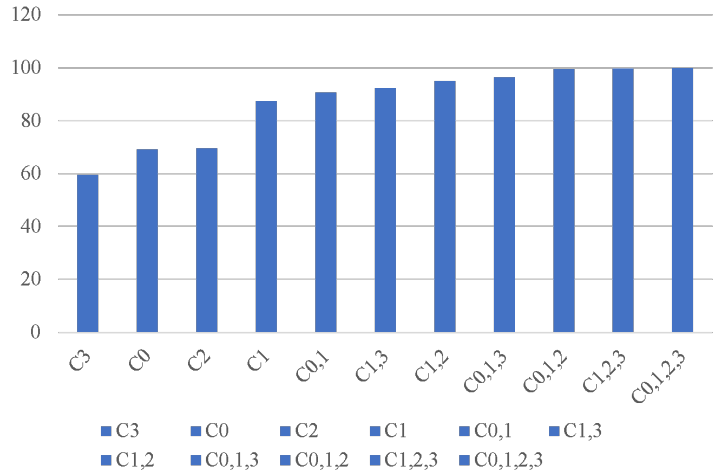

Fig. 14. Relative reconstruction error in the first testing session with different configurations of camera positions, including C0 (top camera), C1(bottom camera), C2 (right /little finger side camera), C3 (left/thumb side camera).

4.2.2 Hand Pose Estimation When the Hand Is Holding Objects. Moving beyond an empty hand, we conducted further evaluation when the hand is holding objects. In this setting, we used the data from the first 32 sessions for training and rest 8 sessions for testing. The background was kept the same for all sessions. The results are presented in Figure 9 (last row). Our system has MAEs of 2.68cm (std ±0.92cm) and 10.37° (std ±3.31°) for joint positions and angles, respectively. Compared to our previous results, the MAE for joint positions and angles increased by a moderate to large margin. We argue that this setting of holding objects in hand is extremely challenging, and thus the results are very encouraging. In fact, to the authors' best knowledge, we are the first to consider estimating hand pose with objects in hand using a wrist-worn wearable system. We have to point out that our study on estimating hand pose with objects in hand is preliminary. For example, only 4 objects were considered in the indoor environment. While our current work only explores the feasibility of reconstructing hand pose with objects in hand, we hope that our study can provide a solid step towards hand pose estimation in the wild. And we plan to further explore this setting in the future.

5 DISCUSSION

5.1 Camera Settings

In the preliminary study with synthetic images, we explored different camera settings. In order to understand how many cameras were needed in the actual device, we conducted another data analysis. In this analysis, we used the data collected in the user study with 11 participants. We used the continuous hand pose data from the first 6 sessions as the training data, and the 7th session as the testing session. We changed the quantity and positions of the cameras we used in the evaluation. We labeled the four cameras used in the study as C0 (top), C1(bottom), C2 (right /little finger side), C3 (left/thumb side). We trained multiple models with different combinations of camera settings, as shown in Figure 14. We calculated the relative reconstruction error in each set of camera setting. We used the evaluation results from all 4 cameras as 100% and compared other reconstruction errors to the original setting. The better result we received, the higher percentage was drawn in the figure. As Figure 14 shows, using only one camera presented the lowest performance. However, C1 is the most informative camera and C3 is the least informative camera among the four cameras. Also, the combination of three cameras (C1, C2, C3) presents a

Fig. 15. Wristband with 4 miniature RGB Cameras used in our pilot study.

similar performance as four cameras. This indicated that in the future, it is possible to remove the top camera from the system but still achieve similar performance.

Fig. 16. Discrete micro-finger poses (12) used in our pilot study (pose recognition task).

5.2 Improvement on Camera Resolution

The thermal cameras used in FingerTrak have several limitations. First, our current model may not work well if the background including objects that present a similar or higher temperature (e.g., sun, heater) as the human body. Second, the resolution of thermal cameras (32×24) is low. Thus, it may not be able to capture subtle changes on the hand shape, which limits the richness of the hand poses that can be estimated. As we have discussed earlier in the paper, there are miniature RGB cameras that satisfy our design needs and provide a higher resolution. Therefore, we built another prototype with 4 miniature RGB cameras (with same position setting) as shown in Figure 15. This new prototype was used to further verify our hypothesis that that the low resolution thermal images may fail to capture subtle differences on the hand shape needed to estimate or distinguish certain micro finger poses. With higher resolution images, it is now possible to capture these subtle changes.

We conducted a pilot study with 10 participants to verify this hypothesis. In this study, the participants were asked to perform a set of hand poses that consists of the thumb touching 12 phalanges, as Figure 16 shown. This set of micro finger poses is arguably considered as the most challenging micro-finger poses to be distinguished as they are highly similar to each other. To evaluate the performance of the wristband with RGB cameras, we asked the 10 participants to wear the wristbands, and performed the 12 finger poses for 18 times in a white background (no segmentation is needed). For each participant, we used the data collected from the first 12 instances as the

training data, and the last 6 instances as the testing data. The model was a deep neural network similar to what we presented in this paper. The only difference was that we changed the input resolution as well as the output of the model (classification). In our pilot study within the research team, we were not able to distinguish these 12 static poses using the thermal images captured from the wristband. The accuracy was under 40%. However, with the data collected from 10 participants using the RGB cameras, we were able to achieve an impressive average accuracy of 92.62%. These preliminary results verified our hypothesis that higher resolution of the images can lead to even more accurate estimation of the hand poses.

What we presented in this paper is a starting point towards the goal of building a practical wrist-mounted device to reliability estimate the complete hand pose. Our study and results showed that it is possible to reconstruct the entire finger pose by observing the hand shape from the wrist. There are multiple ways to further improve the results and apply FingerTrak in real-world scenarios. For example, using a high resolution RGB or depth (RGBD) camera is a natural next step. We have presented preliminary results of using RGB camera in this paper. We plan to experiment with miniature depth cameras. Another possibility is to wait for the technology advancement of thermal cameras, such that we can find miniature thermal cameras with image high resolution. Other improvements include applying deep learning to segment the hand from background, exploring different deep learning architectures and using large-scale synthetically generated data for training.

5.3 Improving Ground Truth Acquisition Method

As we have discussed multiple times in the paper, another major issue that limits the richness and performance of FingerTrak on estimating the hand pose is the stability and accuracy of the ground-truth hand pose, currently provided by Leap Motion. We found Leap Motion cannot recognize complicated hand poses, and sometimes provide unstable ground truth of hand poses, which may influence the training of our model as well as the evaluation of our results

In our preliminary experiment using synthetic images, we were able to accurately estimate very complex hand poses using the 3D hand mesh models, as the parameters allowed us to generate complicated hand postures. Therefore, we believe it is feasible to use FingerTrak to continuously estimate more complicated poses given a device that can provide high quality ground-truth. We considered using a glove to acquire ground-truth. However, we are concerned that the motion-capture glove may change the shape and color of the hand, which may not represent bare-hand performance. We plan to further investigate this issue in the future.

5.4 Influence of Arm Posture and Form Factor Design

In our current user study, the hand poses of a user was collected when the user's arm is on the table. However, in real world applications, the user may want to capture the hand poses under different scenarios with various arm posture. For instance, a user may want to interact with devices while the arm is pointing down the earth. In theory, the arm posture should not influence much of our system, as the positions of the cameras are fixed on the wrist. However, given the current hardware design, when the user moves the arm pointing down to the ground, the weight of the wristband may shift the positions of the cameras a little bit. This shift of camera positions might influence the performance of FingerTrak. One possible solution is to design a more comfortable and form-fitting form factor with lighter weighted cameras, such that the position of the wristband is not easily changed. The other solution is to consider the position change of the wristband when training the model. We will further explore this issue in the next step.

5.5 Re-mounting Error Analysis

One of the possible reasons for the remounting error is the shift of the camera positions after remounting. This is demonstrated in Figure 11, where the captured hand contour of a similar pose had similar shapes yet with a

shift before and after re-mounting. We suspect this is one key factor that causes the decrease of the performance after remounting the hardware. There are a couple of potential solutions to alleviate this issue, including (1) introducing a calibration gesture which helps the system to calculate the shift between different sessions; (2) conducting data augmentation in the training set. For instance, we can normalize and rotate the images to reduce the difference caused by small changes on camera positions; and (3) We can also generate more synthetic training data which includes the images at different angles for the same pose. This will improve the diversity of the data set, which hopefully can help adjust the model to different camera positions. We plan to further explore this in the future.

5.6 Transfer Learning Using Synthetic Images

One challenge of using the current system is that the user has to provide training data first. One solution to this issue, is to take advantage of millions of synthetic hand images that can be produced from our data generation pipeline (as shown in Figure 2) to train a model. Such a model can be trained without using real world data, and deployed to estimate the hand pose in real images with the latest advancement of transfer learning. A major obstacle for training using synthetic data in our current setting is the lack of good simulation of thermal imaging data. While we have used synthetic image data to verify our key idea, the rendered images contains only hand masks and thus are not designed for transfer learning to thermal images. Other challenges of transfer learning include the accurate alignment of camera positions and the user-specific modeling of hands. We believe learning from synthetic data for wearable based hand pose estimation is a promising direction, and plan to explore this direction in the future.

5.7 User-independent Models

We conducted additional experiments using the data from our user study to build user-independent models. Specifically, we performed leave-one-participant-out experiment. Namely, out of the 11 participants, data from 10 participants were used to train the model and the trained model was tested on the other participant's data. This process was repeated 11 times for all participants. Similarly, we report the average MAE of joint positions and joint angles across 4 settings. The results are summarized in Figure 17 (shown as 0% of using a user's training data). Our user-independent models have much worse performance in comparison to their user-dependent versions. For example, our user-independent model achieves a MAE of 8.7 cm in joint positions and a MAE of 14.7° in joint angles, while the user-dependent model has 1.2 cm and 6.46°.

Moreover, we experimented with adding user data during training, leading to leave-partial-participant-out experiment i.e., user-adaptive model. In this setting, we used the data from 10 participants plus a portion of the training data from the other participant for training, and evaluated the trained model on the testing data from the other participant. The results are reported as joint position and angle MAEs in Figure 17. Note that using 0% of user data is equal to leave-one-participant-out. We observe that adding the participant' data can reduce both MAEs by a large margin. However, to achieve a similar performance level of user-dependent models, 80% of the user data has to be added. Moreover, even using all participants' training data, i.e., 100% in Figure 17, the joint position and angle MAEs are 1.29 cm and 7.01° slightly worse than the user-dependent model (1.20 cm and 6.46°).

Finally, we want to point out that we did not expect or claim our current system to be user-independent given a relatively small sample size of data. As the shape and size of hands vary dramatically among different people, the purpose of this paper is to demonstrate a feasibility. How to make the model generalizable among users will be a critical topic for the next step. One possible solution if provided enough resources is to build a model with large enough dataset, that may naturally lead to a model which performs well on different users.

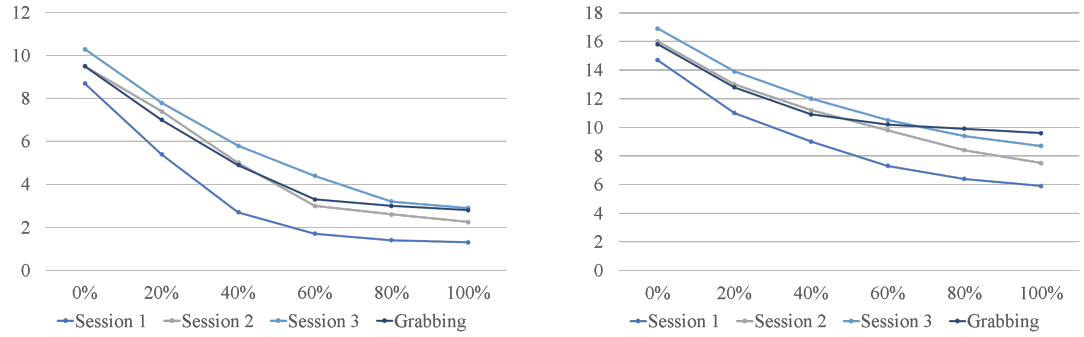

Fig. 17. MAE for joint position (left) and joint angles (right) in leave-one-participant-out experiment. The percent represents the proportion of training data of the other participant used in training the model. 0% means the model is pure leave-one-participant-out model. With the increase of percent, the model is trained with all 10 participants' training data and part of the other one's training data.

5.8 Power Consumption Analysis

The system consists of three parts, cameras to collect images, Raspberry Pi's and a Wi-Fi router to transmit data, and a computing unit to run the model and make predictions. The power consumption of the wearable section of the whole system can be broken down into two parts: on-band cameras and off-band Raspberry Pi's. While the camera can be connected to a micro controller unit (MCU) supplied with VDD=2.6V ~ 5V, the thermal sensor itself is separately supplied from VDD=3.3V or left with no supply (VDD=0V) with the I2C connection running at the supply voltage of the MCU. The current consumption of one camera is under 23mA. Thus, the total on-band power is less than 0.44W. Compared to Apple Watch Series 5, which has 296mAh battery capacity, the on-band part is on a moderate power consumption level. Requiring 5.1V supply, Raspberry Pi 3B+ idles at 435mA with Wi-Fi connected, and 610mA when the CPU is stressed. Without the need to run GUI or GPU and the CPU load is less than 20%, we expect its power consumption to be less than 3.1W. We can further reduce the power consumption by connecting multiple cameras to one Raspberry Pi.

5.9 Applications

Given the encouraging performance on pose recognition of in-session and cross-session, the immediate applications of FingerTrak is to recognize a rich set of hand poses to improve wearable interaction experience. Importantly, the ability of estimating hand poses while holding an object in hand, can potentially enables fine-grained daily activity recognition, such as, the length of reading a book, eating behavior, hand washing behaviors [28], detecting eating moments, and detecting the use of hand-held devices when driving. However, a more thorough study is needed to draw any conclusion. Another interest application of FingerTrak in the future is to replace the glove or controller in Virtual Reality setting, to free the hands in VR interaction. Moreover, FingerTrak can be used to control a robotic hand remotely and thus provide a novel means of human robot interaction where precise human hand manipulation is needed. Last, given enough data and improvement, it is also possible to build a wearable sign language translator using reconstructed finger poses with FingerTrak.

5.10 Limitations and Future Work

The goal of FingerTrak is to demonstrate the feasibility of estimating the complete hand pose from the hand shape captured by cameras on the wrist. The current implementation is not perfect yet, as all prototype does. We have

discussed several other parts that can be improved and explored in the future. For instance, our user-independent model has a much higher error than the user-dependent model which requires the user to provide training data before using the system. Given future large scale datasets, we anticipate that we can potentially improve the performance of user-independent models or develop a user adaptive model with simple calibration. Another example is the study of hand pose estimation with objects in hands. More objects and backgrounds should be considered. And we leave this as part of our future work.

6 CONCLUSION

In this paper, we present FingerTrak, a minimal-obtrusive form-fitting wristband embedded with four miniature thermal cameras, that can continuously track the 3D position of fingers. Our system makes use of a deep model that learns to "stitch" thermal images and to estimate the positions of 20 finger joints in 3D. We demonstrated the feasibility of reconstructing the entire hand pose by only observing a few hand silhouettes from the wrist. Our user study with 11 participants shows that our system can achieve an average angular error of 6.46° when tested under the same background, and 8.06° under a different background. Our results also suggest that our system is able to recover 3D hand pose when the device was re-mounted or when the hand is hold objects. We believe that our study, hardware prototype and results provide a solid step towards hand pose estimation using wearable sensing.

ACKNOWLEDGMENTS

The authors thank Songyun Tao for help with photo capture and video production. The authors also express their gratitude to all reviewers for their valuable feedback, and to all participants for their participation in the user study. Cheng Zhang would like to thank the support provided by Information Science department in Cornell University. Yin Li acknowledges the support provided by the UW-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

REFERENCES

[1] Gilles Bailly, Jörg Müller, Michael Rohs, Daniel Wigdor, and Sven Kratz. 2012. ShoeSense: A New Perspective on Gestural Interaction and Wearable Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. ACM, New York, NY, USA, 1239–1248. https://doi.org/10.1145/2207676.2208576

[2] Simon Baker, Takeo Kanade, et al. 2005. Shape-from-silhouette across time part i: Theory and algorithms. *International Journal of Computer Vision* 62, 3 (2005), 221–247.

[3] Simon Baker, Takeo Kanade, et al. 2005. Shape-from-silhouette across time part ii: Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision* 63, 3 (2005), 225–245.

[4] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. 2012. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision*. Springer, 640–653.

[5] Bruce Guenther Baumgart. 1974. *Geometric modeling for computer vision*. Technical Report. STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

[6] Jean-Baptiste Chossat, Yiwei Tao, Vincent Duchaine, and Yong-Lae Park. 2015. Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2568–2573.

[7] Simone Ciotti, Edoardo Battaglia, Nicola Carbonaro, Antonio Bicchi, Alessandro Tognetti, and Matteo Bianchi. 2016. A synergy-based optimally designed sensing glove for functional grasp recognition. *Sensors* 16, 6 (2016), 811.

[8] James Connolly, Joan Condell, Brendan O'Flynn, Javier Torres Sanchez, and Philip Gardiner. 2017. IMU sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal* 18, 3 (2017), 1273–1281.

[9] Martin de La Gorce, David J Fleet, and Nikos Paragios. 2011. Model-based 3d hand pose estimation from monocular video. *IEEE transactions on pattern analysis and machine intelligence* 33, 9 (2011), 1793–1805.

[10] 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.

[11] Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. 2012. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems* 9, 2 (2012), 36.

[12] Rui Fukui, Masahiko Watanabe, Tomoaki Gyota, Masamichi Shimosaka, and Tomomasa Sato. 2011. Hand shape classification with a wrist contour sensor: development of a prototype device. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 311–314.

[13] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. 2018. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8417–8426.

[14] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D Hand Shape and Pose Estimation from a Single RGB Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10833–10842.

[15] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 41.

[16] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. A stretch-sensing soft glove for interactive hand pose estimation. In *ACM SIGGRAPH 2019 Emerging Technologies*. ACM, 4.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[18] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2019. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 558–567.

[19] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. 2018. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 175–182.

[20] Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer, 492–518.

[21] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

[22] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. 2018. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 118–134.

[23] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*. 303–316.

[24] Frederic Kerber, Michael Puhl, and Antonio Krüger. 2017. User-independent real-time hand gesture recognition based on surface electromyography. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 36.

[25] 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* (Cambridge, Massachusetts, USA) *(UIST '12)*. ACM, New York, NY, USA, 167–176. https://doi.org/10.1145/2380116.2380139

[26] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. 2007. The gesture watch: A wireless contact-free gesture based wrist interface. In *2007 11th IEEE International Symposium on Wearable Computers*. IEEE, 15–22.

[27] Rebecca K Kramer, Carmel Majidi, Ranjana Sahai, and Robert J Wood. 2011. Soft curvature sensors for joint angle proprioception. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1919–1926.

[28] Hong Li, Shishir Chawla, Richard Li, Sumeet Jain, Gregory D Abowd, Thad Starner, Cheng Zhang, and Thomas Plötz. 2018. Wristwash: towards automatic handwashing assessment using a wrist-worn device. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. 132–139.

[29] Tianxing Li, Xi Xiong, Yifei Xie, George Hito, Xing-Dong Yang, and Xia Zhou. 2017. Reconstructing hand poses using visible light. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 71.

[30] Hui Liang, Junsong Yuan, Daniel Thalmann, and Nadia Magnenat Thalmann. 2015. AR in hand: Egocentric palm pose tracking and gesture recognition for augmented reality applications. In *Proceedings of the 23rd ACM international conference on Multimedia*. 743–744.

[31] Bor-Shing Lin, I Lee, Shu-Yu Yang, Yi-Chiang Lo, Junghsi Lee, and Jean-Lon Chen. 2018. Design of an inertial-sensor-based data glove for hand function evaluation. *Sensors* 18, 5 (2018), 1545.

[32] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. *Proceedings of the International Conference on Learning Representations*.

[33] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017. EchoFlex: Hand Gesture Recognition Using Ultrasound Imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1923–1934. https://doi.org/10.1145/3025453.3025807

[34] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect.. In *BmVC*, Vol. 1. 3.

[35] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *2011 International Conference on Computer Vision*. IEEE, 2088–2095.

[36] Corey R Pittman and Joseph J LaViola Jr. 2017. Multiwave: Complex Hand Gesture Recognition Using the Doppler Effect.. In *Graphics Interface*. 97–106.

[37] 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.

[38] 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 *European Conference on Computer Vision*. Springer, 356–371.

[39] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

[40] Kyeongeun Seo and Hyeonjoong Cho. 2014. AirPincher: A HandHeld Device for Recognizing Delicate Mid-air Hand Gestures in Proceedings of UIST. (2014).

[41] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. 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.

[42] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1145–1153.

[43] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. 2018. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 89–98.

[44] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On-and above-skin input sensing through a wearable depth sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3891–3902.

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[46] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence* 20, 12 (1998), 1371–1375.

[47] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*. 945–953.

[48] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 77–89.

[49] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 143.

[50] 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.

[51] Andrew Vardy, John Robinson, and Li-Te Cheng. 1999. The wristcam as input device. In *Digest of Papers. Third International Symposium on Wearable Computers*. IEEE, 199–202.

[52] Chao Xu, Parth H Pathak, and Prasant Mohapatra. 2015. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. ACM, 9–14.

[53] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. 2019. Opisthenar: Hand Poses and Finger Tapping Recognition by Observing Back of Hand Using Embedded Wrist Camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 963–971. https://doi.org/10.1145/3332165.3347867

[54] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, Inan Omer, and D. Abowd Gregory. 2018. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 437.

[55] Yingwei Zhang, Yiqiang Chen, Hanchao Yu, Xiaodong Yang, Wang Lu, and Hong Liu. 2018. Wearing-independent hand gesture recognition method based on EMG armband. *Personal and Ubiquitous Computing* 22, 3 (2018), 511–524.

[56] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 167–173.

[57] Yang Zhang, Robert Xiao, and Chris Harrison. 2016. Advancing hand gesture recognition with high resolution electrical impedance tomography. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 843–850.

[58] Yixin Zhao, Parth H Pathak, Chao Xu, and Prasant Mohapatra. 2015. Finger and hand gesture recognition using smartwatch. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 471–471.

[59] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*. 4903–4911.