

TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images

Karan Ahuja
Carnegie Mellon University
Pittsburgh, USA
kahuja@cs.cmu.edu

Paul Strelci
Department of Computer Science
ETH Zürich
paul.strelci@inf.ethz.ch

Christian Holz
Department of Computer Science
ETH Zürich
christian.holz@inf.ethz.ch

ABSTRACT

Today’s touchscreen devices commonly detect the coordinates of user input using capacitive sensing. Yet, these coordinates are the mere *2D manifestations* of the more complex 3D configuration of the whole hand—a sensation that touchscreen devices so far remain oblivious to. In this work, we introduce the problem of reconstructing a 3D hand skeleton from capacitive images, which encode the sparse observations captured by touch sensors. These low-resolution images represent intensity mappings that are proportional to the distance to the user’s fingers and hands.

We present the first dataset of capacitive images with corresponding depth maps and 3D hand pose coordinates, comprising 65,374 aligned records from 10 participants. We introduce our supervised method TouchPose, which learns a 3D hand model and a corresponding depth map using a cross-modal trained embedding from capacitive images in our dataset. We quantitatively evaluate TouchPose’s accuracy in touch contact classification, depth estimation, and 3D joint reconstruction, showing that our model generalizes to hand poses it has never seen during training and that it can infer joints that lie outside the touch sensor’s volume.

Enabled by TouchPose, we demonstrate a series of interactive apps and novel interactions on multitouch devices. These applications show TouchPose’s versatile capability to serve as a *general-purpose model*, operating independent of use-case, and establishing 3D hand pose as an integral part of the input dictionary for application designers and developers. We also release our dataset, code, and model to enable future work in this domain.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; *Interaction techniques*; Gestural input.

KEYWORDS

Hand pose, depth sensing, capacitive sensing, touchscreens.

ACM Reference Format:

Karan Ahuja, Paul Strelci, and Christian Holz. 2021. TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images. In *The 34th Annual ACM Symposium on User Interface Software and*

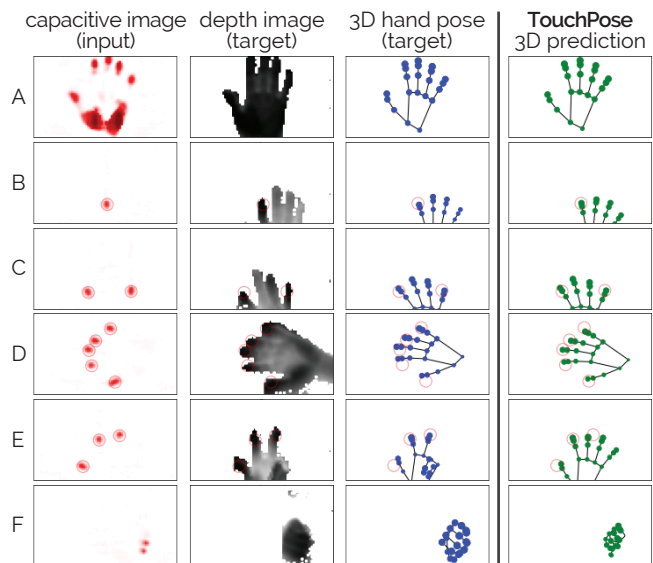


Figure 1: TouchPose takes capacitive images from commodity touchscreen digitizers as input and recovers 3D hand poses. (A–F) Sample predictions by TouchPose on test images from our data corpus of aligned samples from a mutual-capacitance sensor, a depth camera, and a 3D hand pose estimator. TouchPose implements a multi-task scheme to simultaneously predict 3D hand poses and depth images (Fig. 10). Red circles indicate actual contact points for illustration.

Technology (UIST '21), October 10–14, 2021, Virtual Event, USA. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3472749.3474801>

1 INTRODUCTION

Human hand interaction is a cornerstone of how we perceive and manipulate the world. Therefore, much effort in the research communities has gone into capturing and reconstructing hands during such manipulations to extract their spatial configurations with numerous applications in robotics, rehabilitation, Augmented and Virtual Reality. Multi-camera motion capture systems often serve this purpose, but require attaching reflective markers (e.g., Vicon [68]). To forgo the use of markers, previous work has often used optical sensors to directly estimate hand poses from depth [17, 61, 81] and RGB camera images [18, 49].

In Human-Computer Interaction, researchers have used wearable sensors to reconstruct hands during interaction, such as wrist-worn cameras that provide a suitable signal for estimations [33, 78].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
UIST '21, October 10–14, 2021, Virtual Event, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8635-7/21/10.
<https://doi.org/10.1145/3472749.3474801>

Several projects have also attempted to estimate hand poses from much lower-dimensional signals, such as infrared [46] or pressure sensors [12] inside a wrist band. Complementing such indirect methods, gloves have been equipped with sensors to recover the bend of finger joints directly, such as through stretch or bend sensors [9, 19], some of which have led to commercially available products (e.g., CyberGlove [65]).

In addition to involving wearable sensors, researchers have also directly instrumented objects to detect and reconstruct hands as they interact with them. To adapt to smaller and non-planar objects, a frequently used sensing method has been capacitive sensing [22], which is low-cost and scales from small and curved surfaces to larger surface areas, such as tablets and tables [54]. Capacitive sensors can be flexible and integrated into passive objects (e.g., balls [24], computer mice [27, 70]) or attached to the flexible surface of the human body [32, 74] in order to provide information about touch contacts as well as a small amount of hover. A common application of this is estimating the contact between the body and object to reconstruct grasp interaction [76].

In this work, we focus on reconstructing hand poses during interaction with planar surfaces on the devices that we use on a daily basis: the screens of phones, tablets, laptops, and so on. We investigate the problem of recovering 3D joint positions from the spatial intensity map captured by the capacitive sensors that are integrated into such touchscreens. For this, we introduce *TouchPose*, a deep learning model that estimates a fitting 3D hand pose based on a regressor that we devised and trained on the hand pose data captured from 10 participants while they produced touch input on a surface with varying input postures. We demonstrate that our hand regression model benefits from hard-parameter sharing multi-task learning by using a joint embedding space to concurrently estimate the corresponding depth image, the validity of the touch event, as well as the inferred hand pose in real time.

Unlike previous approaches that designed deep networks for *individual* input parameters (e.g., touch classification [10, 38], finger angle [45, 79], inadvertent touch [26, 60]), *TouchPose* is a *general-purpose model* that attempts to recover hand configurations independent of use-case. From each input frame, *TouchPose* produces 3D hand-pose estimates including finger classification, angle, and gesture, offering this information to UI developers for the purpose of interactive applications. Therefore, we see *TouchPose* as a use-case agnostic succession of existing work that trained custom networks with similar complexity on just individual use-cases.

As a side effect and an intermediate result in our processing pipeline, *TouchPose* reconstructs a depth map of hands above the touchscreen surface. Unlike a capacitive image, this recovered depth image *linearly* encodes the distance of hands to-scale, on a per-pixel basis, and with a much larger range than a touchscreen’s few millimeters of sensitivity—much like a depth frame captured with a traditional depth sensor, just with orthogonal projection.

To develop our model, we conducted a data collection as input for *TouchPose*’s training. For this, we constructed an apparatus around a commercial transparent mutual-capacitance panel based on our prior work [64], fitted with a short-range depth sensor and a stereo IR camera for 3D hand pose annotations. The touch panel covered an area close to the size of a sheet of legal paper and the connected digitizer produced capacitive images at 72×41 pixels across an

area of $395 \text{ mm} \times 195 \text{ mm}$. We registered and aligned all recorded samples to obtain capacitive images, high-quality depth images, and 3D hand poses in synchronized pairs and trained *TouchPose* on this data corpus. In a series of experiments, we evaluated our network and show that it can recover hand poses with an average end point error of 21.8 mm per hand joint. The depth maps reconstructed by *TouchPose* are accurate to 22.2 mm on average per pixel.

Because *TouchPose*’s training allows it to operate based on partial observation, i.e., only the parts of the hand that (almost) make contact, *TouchPose* can also infer 3D hand joints that lie *outside* the touch-sensitive area as well as occluded points. We show that *TouchPose*’s reconstruction even generalizes to hand poses and orientations that the network has never seen before. Fig. 1, 9, and 10 illustrate some of the predictions produced by *TouchPose*.

Collectively, our contributions include:

- a learning-based method to estimate 3D hand poses from capacitive images resulting from touch data on sensor surfaces. We train our deep neural network architecture *TouchPose* in a multi-task scheme from 3D hand poses and depth maps captured from a short-range depth camera as ground truth.
- a dataset of capacitive touch images, aligned depth images, and annotated 3D hand poses, captured from 10 participants while touching the surface using various finger combinations, rotations, and angles. The dataset consists of 65,374 samples in total. To enable future research to build on our approach and contribute to this domain, we release our source code, models, and the dataset¹.
- a series of interactive sample applications that are enabled by *TouchPose*’s hand pose regressor to support touch input and novel input techniques.

We believe that our release of *TouchPose*’s dataset, code, and model has the potential to integrate touch contact identities, finger angles, and hand poses as an integral part of future (touch) interaction techniques. This will allow application designers and developers to make use of these dimensions by seamlessly integrating our model into their work.

2 RELATED WORK

Our work sits at the intersection of computer vision, graphics, and human-computer interaction. We start by reviewing the activities in the vision domain on 3D hand-pose estimation and then discuss how they influenced the research on interactive systems, particularly those building on optical and capacitive touch sensing.

2.1 Hand poses from RGB and depth images

Recovering hand pose configurations has been a long-standing problem in computer vision, often addressed using RGB and depth sensors. Li et al.’s survey gives a comprehensive overview of methods and datasets in this context [39].

Camera-based methods typically achieve higher accuracy compared to other sensing mechanisms given their high-resolution capture. Prior approaches using depth sensors include model-based methods that use parameters to encapsulate physical constraints for the validity of results [52, 66, 80]. Alternatively, joint-based

¹*TouchPose*’s source code, model, and dataset: <https://siplab.org/projects/TouchPose>

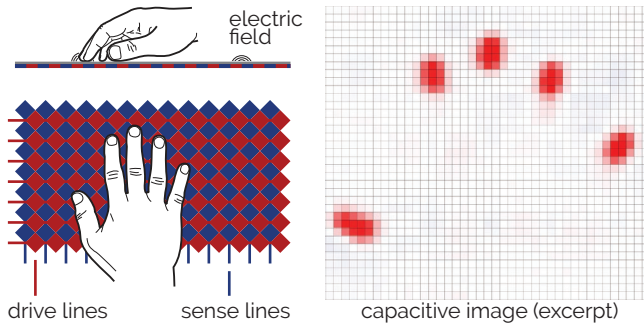


Figure 2: Typical diamond pattern of a mutual-capacitance touch sensor. Approaching fingers couple to the lines and cause a drop in capacitance between them. Right: Resulting capacitive image.

methods directly learn the mapping from depth images to 3D hand pose [13, 51]. In contrast, RGB-based hand pose estimation is more challenging due to depth ambiguity. Yet, following the recent advances in deep learning, many recent projects have estimated 3D hand pose [48, 63] and, more challenging even, 3D hand meshes [18, 83] from single RGB images.

All vision-based efforts have in common that they start from higher-fidelity image data and reduce it to a 3D hand pose. Analogous to frameworks that lift a pose from 2D to 3D [43], wherein multiple plausible solutions exist for a given input, the problem we investigate in this paper is more under-constrained. Our method receives no sensory information from the whole hand and captures only subparts, which a considerably challenging and requires our method to extrapolate from sparse observations.

2.2 Touch imaging and capacitive sensing

Touch-sensing systems, in research as well as in commercial products, commonly rely on optical, pressure, or capacitive sensing. While optical systems have found interest in the research community due to their simple construction [59] and scale [44], mutual-capacitance sensing is underlying virtually all embedded touch technologies today [3], thus widely researched for interactive purposes beyond touch imaging [22]. In here, we focus on methods that investigated touch contact, touch shape, and hover sensing.

In the early 2000s, researchers in Human-Computer Interaction started exploring the benefits of the now common drive and sense line-based matrix pattern for mutual-capacitance touch sensing (Fig. 2). DiamondTouch [14] and SmartSkin [54] showed the capture of 2D touch images on table surfaces and detected individual touches through image processing, touch separation, and tracking. Beyond touch contacts, previous efforts have leveraged the imaging capabilities of capacitive sensors for shape recognition, such as to recognize users (e.g., based on hands [37] or ears [30]), classify touches (e.g., finger vs. palm [36]), detect passive objects (e.g., tangibles [6, 69, 71]), and enable richer hand and arm interaction [16].

Separate from its use in planar touchscreens, a small layer of capacitive sensors can equip otherwise passive objects with input detection for interaction. Past projects have used this advantage for touch and grasp detection. Examples include computer mice

whose surfaces could recognize fingers and palms [27, 70] as well as balls [24] and other tangible objects [76], where camera-based grasp detection is challenging. Capacitive sensors are not limited to rigid surfaces and can add interactive behavior to malleable surfaces using flexible conductors (e.g., in a Project Zanzibar mat [69]).

2.3 3D reconstruction from 2D imprints

Numerous projects have investigated the problem of reconstructing the properties of objects above the touch surface from the contact they make. A repeatedly visited problem is the estimation of finger angles. Through optical sensing, Wang et al. investigated reconstructing the finger yaw angle just from the contact area between the user’s finger and the surface by extracting its principal components [72]. Using a higher-resolution contact sensor, later approaches estimated the finger yaw, pitch, and roll angle from the fingerprint left during touch through template matching [28]. Fiberio performed such tasks in real-time by matching fingerprint minutiae features [29].

Using the much lower-resolution mutual-capacitance sensors in commodity touchscreen devices, researchers have leveraged the fact that they sense a small band of hover to predict finger poses. Examples include Xiao et al.’s [79] and Mayer et al.’s [45] supervised methods to infer finger yaw and pitch angles that operate directly on capacitive images. Closely related to TouchPose, Chung et al. fit a hand model to the touch coordinates using a quadratic encoding [10], but rely solely on the location of touch points and do not make use of the fine-grained, higher dimensional data afforded by the capacitive imprints. As a result, their method is affected by finger ambiguity when less than four fingers are in contact with the surface. Our previous work CapContact is also related, turning capacitive sensors into precise contact sensors [64]. CapContact implements a learning-based upsampling and prediction method, enabling super-resolution capacitive touchscreens that distinguish touching parts of the finger from just hovering parts.

Beyond fingers and hands, an interesting problem in the research community has been the whole-body reconstruction from touch contacts on larger surfaces. On a large optical multi-touch floor, GravitySpace classifies touch contacts into body parts, recognizes users from their shoeprints, estimates their center of gravity, and attempts to reconstruct users’ 3D body poses while interacting [4]. Using much lower-resolution pressure images, Casas et al.’s estimated human pose when lying on a pressure-sensitive bed mattress [5]. In addition to body poses, PressureNet also infers body shapes when lying on a pressure-sensitive imaging surface [11].

Through the use of self-capacitance sensing, prior work has been able to reconstruct input on as well as farther above the surface. Rogers et al. devised a particle filter that simulated finger orientations based on the observations from a low-resolution sensor array to estimate finger motions and angles in 3D [55]. PreTouch fuses such cues from surface and hover input to establish an anticipatory and retroactive hybrid touch model [26], showing a refined estimation of input intentions and locations. Such hover sensing can also be used for detecting hand gestures above the surface [69] or to infer arm orientations and thus user position [82].

3 DATA CAPTURE

We designed our method for supervised multitask learning and now describe our data acquisition method for training and labeling data. The final data corpus we recorded comprises 65,374 pairs of capacitive touch images, 3D hand poses, and depth maps from a total of 10 users for 14 different finger and whole-hand touch poses and gestures. Fig. 4 and 5 show these touch configurations, which were motivated by prior research [15, 79], showing the 3D hand poses from the touch sensor’s perspective. To record samples under controlled conditions, we devised an apparatus and a controlled acquisition procedure.

3.1 Apparatus

Fig. 3 shows the capture rig that we constructed to integrate a mutual-capacitance touch digitizer, a transparent touch panel, a depth camera below the surface, and a stereo camera above the touch surface based on our prior work [64]. A 39.6 cm capacitive Crystal Touch panel (Ocular Touch, Dallas, TX) is at the center of our rig, mounted at 1.15 m above the ground inside the aluminium profile construction (item Industrietechnik, Solingen, Germany). The panel was made from transparent glass and bonded ITO-based semiconductive traces in the common diamond pattern through drive and sense lines [62]. The touch-sensitive area on the panel was 345 mm × 195 mm at 72 × 41 lines with a industry-standard pitch. The panel connected to the digitizer (MXT2954T2 touchscreen controller, Microchip, Chandler, AZ), which recorded 16-bit gray-level capacitive images.

Fig. 2 shows the sensing principle of the touch sensor. As a finger approaches the surface, the digitizer registers a drop in mutual capacitance between the respective drive and sense lines. The drop is proportional to the distance and impacted by other factors such as finger properties (e.g., size, skin and tissue characteristics) and grounding (e.g., gloves, shoe materials, floor properties). Through factory calibration and built-in algorithms, touch digitizers compensate for these and other factors (e.g., electromagnetic noise).

For ground-truth labels, the rig included a Leap Motion stereo IR camera (Ultraleap [67]), running Orion 4.1, which is widely used for 3D hand pose tracking. Leap’s estimates have sub-millimeter accuracy under dynamic movements [73], which has made it a popular sensor for hand pose and gesture recognition [20, 42, 53, 58]. Its single vantage point and ease of setup has made it prevalent as a ground truth acquisition device for human-computer interaction [31, 40] and computer-vision based hand pose datasets [21]. We chose the Leap Motion camera over marker-based tracking (e.g., Vicon [68]) because of our close-range setup, small interaction area, and occlusion-free field of view from the sensor. In addition, retroreflective markers needed for motion-capture tracking would have restricted participants from freely pressing their hands onto the touch surface at different orientations (especially for gestures that involve the palm as we explain later).

The Leap returned a 3D hand skeleton with 21 joints: 4 for each of the 5 fingers and 1 for the wrist. For depth image collection, the rig integrated an Azure Kinect [2] (Microsoft, Redmond, WA), which is powered through Kinect DK 1.4.1. The camera produces a systemic error of less than 11 mm in short-distance mode [47]. As the Leap provides us only with hand skeletons, we collected depth data to

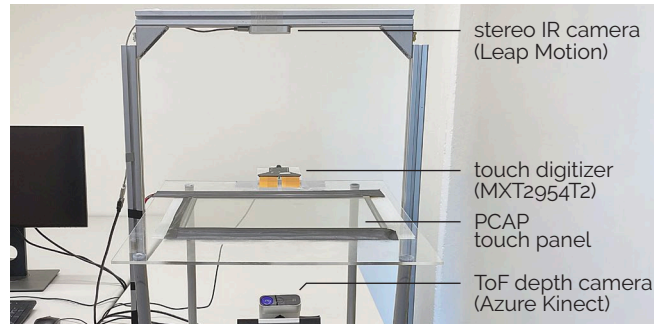


Figure 3: Data capture rig with mutual-capacitance touch panel, 16-bit touch digitizer, and cameras to record ground-truth data.

aid in resolving the ambiguity of finger width (and, by extension, exact finger-touch points). The capture of depth for our dataset was also motivated by potential future hand pose recognizers that could replace the skeletons from the Leap’s low-resolution recording by analyzing the more accurate and higher-resolution depth frames.

We developed a tool to control all capture devices and to interface with all sensors. Our software included calibration controls to register the depth camera and the Leap Motion to the same reference system, which we conducted before each capture. In detail, this included a 4 (corner) point calibration. For depth, the experimenter placed a sheet of paper on the touch surface and retrieved the Kinect-provided 3D coordinates for each corner. For Leap, the experimenter touched each corner with the index fingertip. The calibration then resulted from the homography between the corresponding 3D coordinates and surface coordinates. We transformed and rendered all depth images through orthographic projection from the sensor’s point of view. The Leap and Kinect were placed to not be in line of sight of each other and hence did not interfere.

Our software tool also provided controls for additional settings of the touch digitizer (e.g., to disable all chip-implemented dynamic noise control, filtering, and adaptive processing) and camera settings (e.g., depth granularity and frame rates). Finally, our tool drove all three sensors in unison and ensured the synchronicity of captured frames by recording the temporally closest samples from each sensor. The tool collected paired samples at 9 fps on average.

3.2 Participants

10 participants were recruited for data collection (two female, eight male, ages 22–35 years, mean = 28 years). Participants were recruited broadly to cover anatomic differences in hand characteristics, which covered various ranges with regard to the length of the middle finger (75–92 mm, mean = 83 mm), the length of the right hand (176–209 mm, mean = 190 mm), and the width of the middle finger at the distal joint (16–20 mm, mean = 18.5 mm).

3.3 Data acquisition procedure

To capture representative data, participants produced different touch gestures using their right hands as shown in Fig. 4 and 5. For each combination of fingers, participants input a sequence of touch events following instructions: place the finger(s)/whole hand

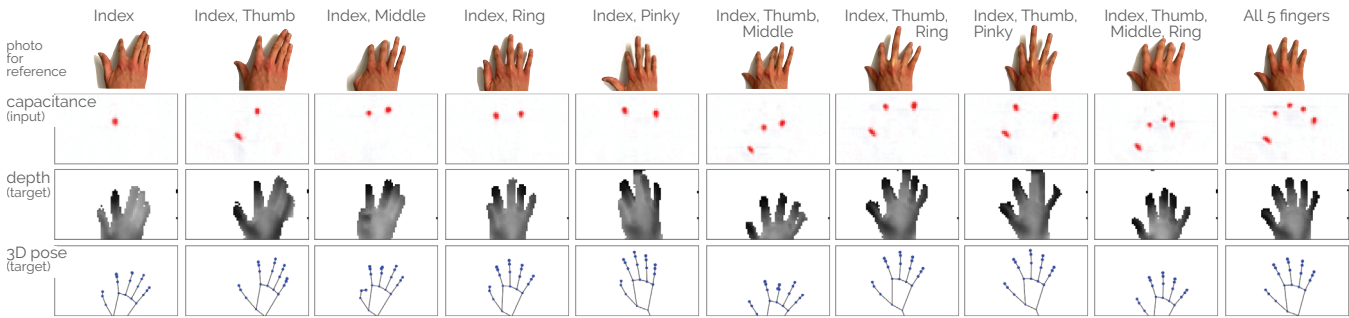


Figure 4: The 10 finger poses in our dataset. During each of the conditions (column), the labeled finger(s) were in contact with the surface. Note: The RGB pictures are for reference only and taken from above, while the depth images capture the hand from below. The 3D hand poses are visualized from the touch sensors perspective.

onto the panel, idle for a second. Lift the finger(s), place them back onto the panel, move them towards to the bottom. Lift and place back down. Push them up, before finally releasing the touch(es). Afterwards, participants rotated their arms 90° counter-clockwise (pointing their elbows to the right) and repeated the same sequence left to right instead of top-down. This procedure enabled our dataset to account for both landscape and portrait mode input.

Throughout the collection, participants were encouraged to vary their finger angles freely and also to touch different locations on the panel during each trial, but they received no definite instructions. Participants performed the interactions at their own pace, resulting in different velocities and motion profiles.

Participants repeated the procedure described above three times, resulting in three sessions. Each block encapsulated 2 starting yaw angles × (10 finger combinations + 4 whole-hand poses) with short breaks in between. The whole data collection lasted for just over an hour for each participant.

3.4 Data pre-processing and filtering

The software logged the frames from each sensor in a shared coordinate system due to the calibration prior to each collection, whose origin is at the top-left corner of the panel. The resolution of the depth image was adjusted to match that of the capacitive images. Each pixel in the depth image resulted from the mean z distance

across all depth vertices that mapped to the pixel after x–y projection. Additionally, the capacitive image was pruned from negative values, which stem from interconnected electrode drive and sense line patterns in the case of multi-touch inputs.

In total, our experiment captured ~150,000 frames. To ensure data quality, we scrutinized all recorded frames and excluded unsuitable frames to prevent them from entering our procedure. First, all samples without any 3D joint information present over the touch panel were removed. This included 52% of the data that did not contain any touches (empty capacitive images) and 4% of the data that had erroneous Leap data (no 3D joints present over the touch panel). To ensure data consistency, we discarded the few samples (~1%) where the distance between the 3D fingertips (provided by the Leap) and the centroids of touches in the capacitive image was greater than a threshold.

Data filtering left 65,374 samples in the final corpus from a total of 10 users for 14 different finger and whole-hand touch poses and gestures (Fig. 4 and 5). Each pose is represented with 3030 to 5875 samples (mean = 4670, SD = 700). The minor imbalances between the number of images can be attributed to the different speeds with which participants completed the different touch events. Fig. 6 shows the histogram of the joint-angle distribution of the filtered data. In summary, across all joints our dataset had a mean yaw and pitch of 17° (SD=69°) and 38° (SD=23°), respectively.

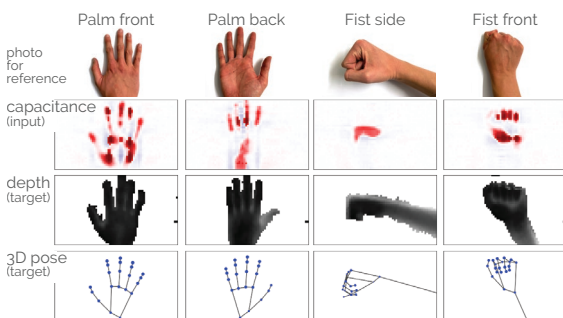


Figure 5: The 4 whole-hand poses in our dataset. Each column shows a condition. RGB pictures are for reference only.

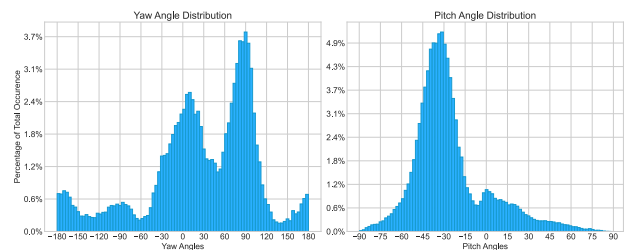


Figure 6: Dataset joint-angle distributions across all fingers for yaw (left) and pitch (right), respectively.

4 METHOD

Estimating hand pose from capacitive images is a challenging problem as there are multiple plausible solutions, especially for the fingers that are not in contact with the surface (Fig. 7). To constrain our search space and motivate the architecture of our network, we made a few observations during the data collection. First, whole hand-based events (Fig. 5) have a hand pose profile that significantly differs from fingertip-based touch events, leading to greatly different capacitive images (Fig. 4). Second, the fingertips of estimated hand poses may not correspond to the locations recorded by the touch sensor, which can be attributed to the fact that a finger is volumetric (17.5 mm diameter on average [7]). The touch sensor registers the part of the finger that makes contact with the surface, which is offset from tip’s 3D joint position. Analyzing the ground-truth data, we found this offset to be 9.32 mm on average, resulting from participants’ average finger width of 18.5 mm (Section 3.2). In such cases, the depth image provides a holistic capture of the whole hand shape in its point cloud.

4.1 Multi-task Convolution Neural Network

We propose a learning-based model that estimates the 3D pose of a hand H touching a sensor surface based on the corresponding capacitive image C as input. While it is intuitive that there exist multiple possible hand poses for a given input—especially in the case of few fingers touching the screen (Fig. 7)—our method aims to recover the best fitting hand pose based on the collected data in a least-square sense.

Motivated by the aforementioned observations, we add two auxiliary tasks to improve the generalization of our model. We train our model to not only predict the hand pose H but also the corresponding depth image D as well as the likelihood that the capacitive image resulted from a fingertip-based interaction. With these changes, the network now shares a common embedding for all three tasks, following an approach that is commonly known as hard-parameter sharing multi-task learning [57].

We model the capacitive frame C and the depth image D as real-valued tensors of dimensions $41 \times 72 \times 1$. The capacitive image input frames are normalized between 0 and 1. The hand pose H consists of 21 joints, each represented by three Cartesian coordinates, and modeled as a real-valued tensor of size 21×3 . The likelihood estimate TC is a real-valued scalar, either 0 for fingertip events or 1 representing whole-hand events.

4.1.1 Architecture. As shown in Fig. 8, our model consists of a UNet-shaped architecture [56] that encodes the capacitive image into a lower-dimensional embedding space via three downsampling blocks, each consisting of two 2D-convolutional layers with a kernel



Figure 7: The same capacitive image (left) can result from different hand poses due to variations in orientation of fingers that are not touching the screen. Here, only index and middle fingers are touching the screen.

size of 3 followed by a max pooling layer. Another convolutional layer maps the resulting representations to the embedding space of dimensions $5 \times 9 \times 256$.

The expansive part of the network retrieves a depth image from the extracted embedding. It comprises three convolutional blocks that use nearest-neighbor interpolation for upsampling. A final convolutional layer reduces the channels of the output to a single dimension. After each upsampling layer, we concatenate the corresponding feature maps from the contractive part—extracted before the max pooling layer—to the upsampled representations. To obtain a depth image of dimension $41 \times 72 \times 1$, we add another row to the output of the last upsampling layer using symmetric padding. For the estimation of the 3D hand pose, the embeddings from the bottleneck of the UNet are flattened and fed into two dense layers. Based on the resulting representations, a linear layer predicts the coordinates of the 21 hand joints and another dense layer followed by a sigmoid activation function estimates the likelihood that the capacitive image represents a fingertip event.

4.1.2 Loss function. Our loss function,

$$L = L_H + \alpha L_D + \beta L_{TC} \quad (1)$$

consists of three terms—one for each predicted output—that are weighted by hyperparameters α and β .

We use mean squared error to calculate the loss for the predicted hand pose L_H ,

$$L_H = \frac{1}{J \times 3} \sum_{i=1}^J j_i - \hat{j}_i^2 \quad (2)$$

where J is the number of predicted joints, and j_i and \hat{j}_i are the coordinate vectors of the actual and predicted i -th joint respectively.

L_D is the mean squared error between the ground truth depth image D and the predicted depth image \hat{D} ,

$$L_D = \frac{1}{P} \|D - \hat{D}\|_2^2 \quad (3)$$

where P is the number of pixels in the image.

Finally, we penalize the output of the touch classification network path \hat{t} by comparing it with the ground-truth label t , classifying a touch as whole-hand or fingertip event, using a binary cross entropy loss L_{TC} ,

$$L_{TC} = (-t \log(\hat{t}) - (1-t) \log(1-\hat{t})). \quad (4)$$

4.1.3 Network Training. Our multi-task CNN has 4,937,345 trainable parameters. During training, we use a batch size of 32 and update the weights using the Adam optimizer [34] with a learning rate of 0.001 and a decay of 5×10^{-6} . For our loss term (Equation 1), we use $\alpha = 0.25$ and $\beta = 0.2$. We train our model for 200 epochs on an NVIDIA Titan V GPU which takes approximately 4.2 hours. The network is implemented in Tensorflow and incurs an inference latency of 24 ms per input.

4.1.4 Inverse Kinematics and Hand Mesh Estimation. As a final step, we pass the 3D hand skeleton estimated by our multi-task CNN to an inverse kinematic solver. This helps correct unnatural hand poses and also facilitates the rigging of a valid hand mesh. We make use of *IKNet* [83] to estimate the hand mesh in a MANO hand model, as shown in Fig. 9. This helps achieve a more natural look,

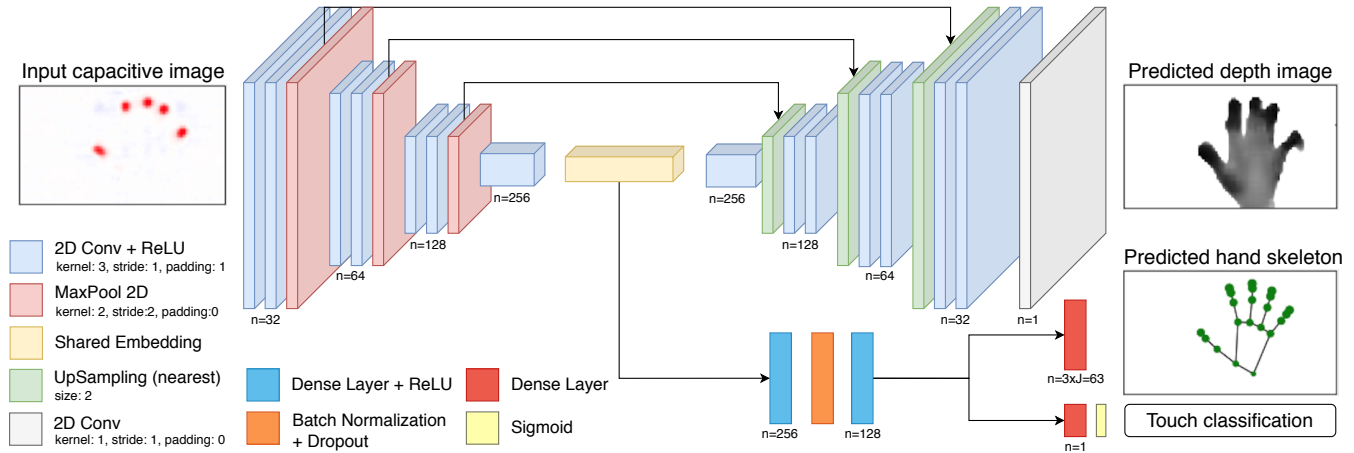


Figure 8: Overview of our architecture. It is a multitask learning framework, which takes in capacitive image as input and predicts the depth image, 3D hand pose and touch classification for it.

reject outlier poses and allows us to continue animating the hands through sparse input periods.

4.2 Baseline: Nearest Neighbor + Inverse Kinematics

To quantify the efficacy of our multi-task CNN, we created a simple baseline for pose estimation. Prior works have used Inverse Kinematics (IK) as a baseline [1, 50] using a priori fingertip location. We first reduce the dimensionality of capacitive images by flattening them and deriving the 128 principal components [77] to represent each image, similar to Chung et al. [10]. We do this for all the images in our “matching” dataset, arriving at pairs of 128-dimensional vectors and their corresponding 3D hand poses. We experimented with different numbers of principal components and achieved similar results for greater than 100 principal components.

For an incoming test candidate capacitive image, we first transform it into a 128-dimensional vector as described above and then run a k nearest neighbor search to find the top 25 matches in our dataset using an Euclidean distance metric. We then take the 25 matched 3D hand poses and average them, each weighted inversely to its distance error. We pass the resulting 3D hand pose prediction

through our IK pipeline as discussed in Section 4.1.4 to output the final pose estimation.

5 EVALUATION

We now first describe our training and testing protocols and show the results on our collected dataset. Our ablation study then evaluates our design choices. We outline three evaluation protocols to test the efficacy of our hand pose estimation models.

- **Protocol 1:** We employ a 3-fold ‘leave-one-session-out’ cross-validation wherein each fold consists of training data from two sessions and is tested on the remaining one. This is used to study the per-user effects (such as hand scale, geometry, etc.) on the accuracy.
- **Protocol 2:** We use a 10-fold ‘leave-one-person-out’ cross-validation wherein each fold consists of data from nine participants for training and the data from the remaining participant for testing. This introduces cross-subject variation in hand scales and pose styles.
- **Protocol 3:** We use a 14-fold ‘leave-one-gesture-out’ cross-validation wherein each fold consists of training data from 13 hand gestures and the remaining fold has a hand gesture that it has never seen before. This tests the generalizability of our model to unseen hand poses.

Unlike prior hand tracking methods that leverage global alignment, scaling or root-centric error calculations, we calculate all our poses and errors in a touch-centric coordinate system. Herein, the upper left corner of the touch panel is the origin. This helps to account for global errors in orientations and different hand-scales without training custom models or calibrating for bone length.

6 RESULTS

Table 1 summarizes the performance of TouchPose under the different evaluation protocols. As our system outputs the full hand pose, we can operationalize different factors motivated by prior research and perform an in-depth analysis.

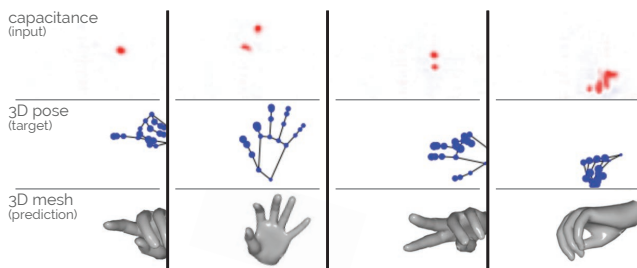


Figure 9: Top: Raw capacitive input images. Middle: Ground-truth hand poses for comparison. Bottom: Hand meshes rendered using skeletons that were estimated by TouchPose.

Variants (EP)	FingerID(%)	Yaw Err (°)	Pitch Err (°)	EPE (mm)	EPE_v (mm)	AUC	Depth Err (mm)
TouchPose (1)	91.1 (SD=3.4)	10.4 (SD=1.2)	8.8 (SD=0.6)	19.6 (SD=2.7)	13.8 (SD=1.9)	0.85 (SD=0.05)	20.7 (SD=1.6)
TouchPose (2)	88.0 (SD=10.4)	11.4 (SD=3.2)	9.9 (SD=2.5)	21.8 (SD=7.1)	15.8 (SD=3.8)	0.83 (SD=0.10)	22.2 (SD=3.3)
TouchPose (3)	83.1 (SD=13.3)	11.1 (SD=2.4)	9.6 (SD=1.5)	29.2 (SD=17.5)	15.4 (SD=4.6)	0.70 (SD=0.28)	24.9 (SD=7.1)
TouchPose (3*)	83.1 (SD=13.3)	11.1 (SD=2.4)	9.6 (SD=1.5)	18.4 (SD=6.3)	15.4 (SD=4.6)	0.86 (SD=0.10)	21.1 (SD=4.7)
Baseline (1)	83.4 (SD=5.2)	16.5 (SD=2.8)	11.5 (SD=1.2)	24.3 (SD=5.2)	16.8 (SD=3.3)	0.82 (SD=0.09)	NA
Baseline (2)	76.2 (SD=11.3)	18.7 (SD=4.3)	13.2 (SD=2.7)	32.6 (SD=8.4)	29.9 (SD=6.9)	0.71 (SD=0.13)	NA
Baseline (3)	69.9 (SD=12.1)	21.8 (SD=6.1)	15.8 (SD=4.5)	40.2 (SD=18.6)	32.7 (SD=15.3)	0.61 (SD=0.14)	NA
Baseline (3*)	69.9 (SD=12.1)	21.8 (SD=6.1)	15.8 (SD=4.5)	36.2 (SD=16.1)	32.7 (SD=15.3)	0.69 (SD=0.11)	NA

Table 1: Quantitative results of different TouchPose variants across the evaluation metrics and protocols. All numbers reported are in mm. * denotes accuracy calculated on only fingertip events. Here, FingerID denotes the finger classification accuracies and EP stands for Evaluation Protocol.

6.1 Finger classification during touch

We first evaluate the efficacy of our system for classifying the finger that each fingertip interacting with the touchscreen belongs to. For this, we only look at our 10 fingertip-based interactions (Fig. 4) and the fingers touching the screen. Since our model predicts hand pose rather than finger ID, we map each touch point to the closest fingertip by finding the one that has the lowest Euclidean distance to it, respectively, with a one-to-one mapping between the two.

Under the cross-session cross-validation (Evaluation Protocol 1), TouchPose has the highest finger classification accuracy of 91.1% (SD = 3.4%), which falls to 88.0% (SD = 10.4%) for the cross-person scenario (evaluation protocol 2). Under Evaluation Protocol 3, where the model is tested on finger combinations it has not seen before, we found a mean accuracy of 83.1% (SD = 13.3%). This compares favorably to our baseline (Section 4.2), which has a finger classification accuracy of 83.4%, 76.2%, and 69.9% for Evaluation Protocols 1, 2, and 3, respectively.

6.2 Finger angle estimation during touch

Similar to Section 6.1, we evaluate the mean absolute error (MAE) in the yaw and pitch for the 10 fingertip-based events as well as only for fingers that are touching the surface. Table 1 shows a detailed breakdown of all results. In summary, averaged across all evaluation protocols and fingers, TouchPose consistently performs better than the baseline with a MAE of 11.0° (vs. 19.0°) for yaw and a MAE of 9.4° (vs. 13.5°) for pitch.

6.3 3D hand pose estimation error

To test the efficacy of our 3D pose pipeline, we make use of the following hand pose evaluation metrics:

- **End-point-error (EPE):** This commonly used metric for 3D hand pose estimation is the mean Euclidean error between all the joints (= 21) of the Leap and predicted hand pose.
- **End-point-error of visible joints (EPE_v):** Similar to EPE, but only computed for the finger joints touching the screen. For example, if only the index and ring finger are touching the screen, we will only compute the EPE for all the joints along those fingers. Hence, its calculated for fingertip events only (i.e., the 10 poses in Fig. 4). The reason is that many plausible hand poses exist for fingers that are not in contact

with the surface and that errors in the ones that make contact should be scrutinized in isolation.

- **AUC under PCK:** Area under the curve (AUC), which represents the percentage of correct 3D keypoints (PCK) of which the Euclidean error is below a threshold t , where t ranges from 20 mm to 50 mm.

Similar to our previous finger classification and angular error-based results, cross-session cross-validation (EPE mean=19.6 mm, SD=2.7 mm) produces much lower errors than cross-person cross-validation (EPE mean=21.8 mm, SD=7.1 mm), showcasing that per-user data aids in performance. The higher standard deviation when validating cross-person can be attributed to participants’ differing styles how they touched the surface, especially the pose of fingers that were not in contact (Fig. 7)—some participants preferred tucking in hovering fingers, while others kept them stretched out.

The error of the visible joints (EPE_v) is more uniform across participants, with a mean EPE_v of 15.8 mm (SD=3.8 mm). The PCK curves of TouchPose across different evaluation protocols are shown in Fig. 11. For our auxiliary tasks, our model achieves a touch classification accuracy of 99.58% and our depth reconstruction mean absolute error of 22.2 mm under Evaluation Protocol 2.

Sample hand pose predictions on our test set under Evaluation Protocol 2 are shown in Fig. 1. TouchPose can adapt to different

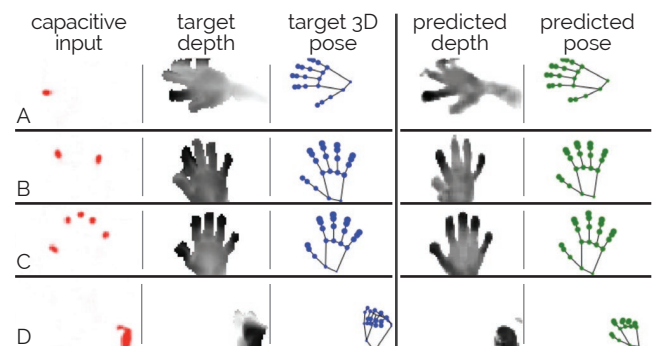


Figure 10: Sample predictions from TouchPose on hand poses that it has not been trained on (Evaluation Protocol 3): (A) index finger, (B) index and pinky, (C) all five fingers, (D) side fist.

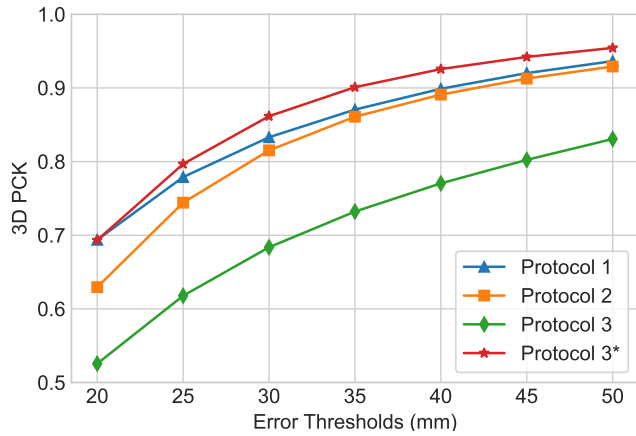


Figure 11: The PCK (percentage of correct 3D keypoints) curve of TouchPose across different evaluation protocols. * denotes accuracy calculated on only fingertip based touch events.

hand orientations and it can also estimate joints that lie outside the touch sensor area. Moreover, it can also account for the offset between Leap-reported 3D coordinates and detected contact position centroids, and further minimize this in its output as shown in Fig. 1D. In some cases, the predicted hand pose is erroneous for some joints. Fig. 1E shows an example where the ring and pinky finger do not make contact with the surface and are incorrectly predicted. Fig. 1F shows an example where the scale of the hand predicted by the model is smaller than the target hand.

Evaluation Protocol 3 proves to be the most challenging for TouchPose, with the model having its highest mean EPE of 29.2 mm (see Fig. 10 for sample predictions). Upon further analysis, we noticed that most of the errors were in whole-hand based touch events. As the capacitive images as well as the skeletons from each whole-hand poses differ greatly from the rest of the samples (see Fig. 5), predicting their pose without any prior knowledge is indeed a challenging task for which the network fails to generalize well. This can be seen in Fig. 10D, where the model fails to reconstruct a *first side* hand pose. We thus tested the ‘leave-one-gesture-out’ protocol by validating solely on unseen fingertip based touch events (*Protocol 3**) which turn up in greater variety within the dataset. For these touch events, our model achieves a mean EPE of 18.4 mm. Visualizing the predicted depth image and the predicted hand pose, we find that our model successfully assigns the capacitive touches to the respective fingers (Fig. 10A&B). Inspecting the predicted depth images, we can also see that TouchPose learned the relationship between the proximity of an object and the resulting change in capacitance as recorded by the touch panel.

In all cases, as expected the EPE_o is smaller than the EPE, showcasing fingers that touch the screen are correctly predicted and less ambiguous than the ones that do not. The EPE also decreases proportionally to the number of fingertips touching the screen, decreasing from 18.3 mm for one finger to 12.0 mm for four fingers touching the screen (reduction of 6.3 mm). This was expected—at the moment a user touches the surface, the degrees of freedom of

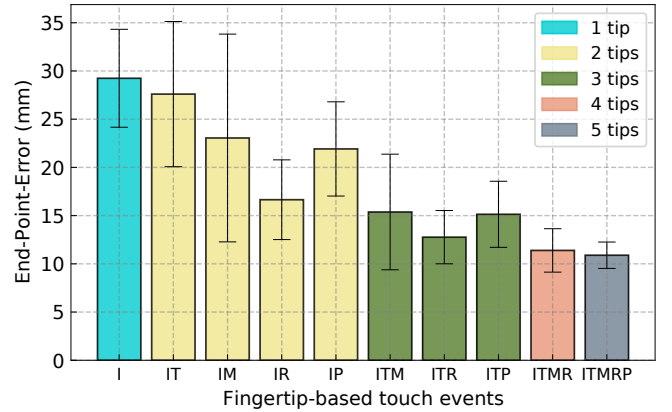


Figure 12: EPE vs. different fingertip touch events under Evaluation Protocol 3. Letters denote the touching fingertips—I: Index, T: Thumb, M: Middle, R: Ring, and P: Pinky. The legend denotes the number of fingertips touching the screen.

the hand pose decrease (the touch locations limit the 3 degrees of freedom for the respective end effectors). Therefore, as the fingers make contact with the surface, the more constrained the search for the whole hand. This is also evident under evaluation protocol 3, where we observe that the model’s predictions become sharper (e.g., comparing the depth images for Fig. 10A,B,C) as well as more accurate as the number of fingers on the surface increases and thus the degrees of freedom for the hand pose decrease. This relationship between the EPE and number of fingertips touching the screen is depicted in Fig. 12.

6.4 Ablation Study

We evaluated the key design choices we made for TouchPose during development in an ablation study. In particular, we tested the performance of our model without the depth reconstruction loss and without the touch-based classification loss. All the results are for Evaluation Protocol 2. Our touch classification is therefore suitable for quickly rejecting inadvertent whole-hand events, which require no processing. In particular, predicting the auxiliary task of classifying the touch event between whole-hand and fingertip-based events (by adding L_{TC}) helps decrease the EPE by 0.6 mm. Without L_D and L_{TC} , the AUC falls by 0.02, from 0.83 to 0.81. Correspondingly, the finger classification accuracy decreases by 3.4% and the angular errors of the finger touch points increase by 4.5%.

While the decrease in EPE (2.5%) appears to be modest, the integration of depth notably decreases the distance between the hand pose and the touch points from a mean error of 9.3 mm in the target samples to 8.0 mm in the predictions of the model (a decrease of 14%). Perceptually, this leads to a more stable output and brings our predicted pose closer to the touch points on the screen (Fig. 1D).

Apart from the multiple losses, the inverse kinematic module also helps decrease the EPE by 2.5% and produces hand poses that are temporarily more coherent (less jitter between subsequent predictions). We also experimented with α and β combinations in the

range of 0.1 to 0.7 (see Equation 1) and noticed that EPE varied by 7% for different combinations.

6.5 Comparison with prior work

To the best of our knowledge, no prior research has investigated deriving full-hand 3D poses using capacitive touchscreen data. This makes directly comparing TouchPose with prior work difficult. State of the art hand pose estimation methods that take monocular RGB [83] and depth images [23] as input have achieved AUC of PCK of up to 0.948. For a comprehensive guide, we direct the reader to Chatzis et al.’s overview of learning-based methods [8]. The best case AUC of PCK produced by TouchPose on the whole dataset is 0.85 under Evaluation Protocol 1 (see Table 1).

DeepFisheye [53] uses a fisheye camera mounted to the touchscreen to predict which fingertip is touching the screen, achieving a mean Euclidean error of 20.1 mm. Compared in similar terms, TouchPose has an average fingertip Euclidean error of 20.4 mm (under Evaluation Protocol 2), using no external sensors and only the touch sensor’s data as input.

In terms of finger labeling and identification, Chung et al. make use of a quadratic encoding to represent the different touch locations and reconstructs the user’s hand pose from it [10]. Their system has an accuracy of 55.6%, 70.7%, 79.4% and 99.8% for identifying the fingers when two, three, four and five fingers are touching the surface, respectively. TouchPose achieves similar accuracy levels for the 5 finger scenario (99.7%) and produces considerably higher accuracies for two (74.7%), three (93.1%) and four fingers (96.1%) touching the surface under Evaluation Protocol 3 (Section 6.1).

For distinguishing input events, PalmTouch can distinguish between finger and palm touch with an accuracy of 99.53% [36]. TouchPose accomplishes the same with a comparable accuracy of 99.58% (F_1 score = 99.13%) as one of its auxiliary tasks. For finger angle estimation, Mayer et al.’s CNN achieves a Mean Absolute Error (MAE) of 9.9° and 18.3° for pitch and yaw of the index finger, respectively. TouchPose has a similar error for pitch angle estimation, but a much lower MAE for yaw, ranging from 10.4° to 11.4° across different evaluation protocols (see Table 1).

We attribute TouchPose’s lower error to our multitask learning framework that benefits from predicting the holistic hand pose. We note that these comparisons are provided for guidance and future reference, as all these methods were tested on different datasets and were designed with different applications in mind. The release of our dataset and model should therefore contribute to comparisons with future methods.

7 DEMONSTRATION APPLICATIONS

Prior research has demonstrated many applications of sensing the hands from capacitive images, such as grasp recognition and tool manipulation [10, 25, 35], 3D interaction with content [75, 79] and character animation [41]. TouchPose’s 3D hand pose can serve as a superset for these applications. Thus, it can be used as a general-purpose model that performs independent of use-case, providing hand poses (and by extension finger ID, angle, gesture) to UI developers. We showcase a few of the many possible interactive applications TouchPose can enable as shown in Fig. 13.

Using the capacitive touchscreen on an Android phone (Honor 7X, 15 cm display, capacitive input offloaded and processed following prior approaches [37]), we illustrate examples that make use of TouchPose’s unique capability to recover hand and finger poses as well as implicitly identify the part of the hand that is in contact with the screen. Fig. 13A shows a drawing app using this to select the input tool and parameterize the stroke width according to the finger angle in pitch and yaw, all with a single input with feedback displayed in real-time. TouchPose can also power applications to afford quicker input to UI controls, such as the slider in Fig. 13B that adjusts based on finger pitch, making TouchPose a suitable processing layer for modeling applications on mobile devices where surface area is scarce.

A final app, which could also prove useful most immediately, processes input events into parts of the hand and finger to assess their suitability for the given input operation. Unlike current text editing apps on touch devices, the app shown in Fig. 13C incorporates TouchPose for transparent processing of input events, letting (C1) valid type events triggered by fingers pass while rejecting inadvertent input caused by (C2) invalid parts of the finger or palms and (C3) sides of the hand. We see particular potential in this app, as touchscreens lack the capability of distinguishing between touch types. This problem may be even more severe on larger-screen devices such as tablets or tables, where users tend to rest their palms or wrists while providing input or while writing with a stylus to support accurate input.

In addition to the demonstrations we implemented, we also envision apps in Augmented Reality where hands serve as controllers to manipulate and interact with 3D virtual objects, therefore providing a means for immersive user interactions.

While we did not explicitly validate TouchPose’s accuracy on a variety of multitouch devices that operate on different resolutions and aspect ratios, we expect TouchPose to generalize beyond the touch sensors we tested on in this work. Despite differences in configurations, modern touchscreen devices use similar sensing pitches and since our implementation is based on a convolutional architecture, TouchPose should be able to accommodate other devices. We also expect that the small differences in sensor configurations that do exist are accounted for through variations in participants’ finger sizes in our data collection.

8 LIMITATIONS AND FUTURE WORK

While TouchPose demonstrates the feasibility of our approach, there are several limitations that will benefit from further refinement. First is the dataset itself. While we collected a large corpus of data, it can be extended to include more varying poses (e.g., fingertip nails, knuckles, side-hand poses, etc). Furthermore, TouchPose is currently trained on the right hand, which could be extended for left hands by flipping the input. In its current form, it cannot handle multiple hands interacting simultaneously. In the future, we plan to record additional data to account for this.

We also acknowledge that our current implementation cannot disambiguate finger classes in single-touch events. TouchPose recovers “partial” hand poses, as there is inherent ambiguity in missing contacts. However, we note that as more fingers make contact, ambiguity drops and finger identification confidence improves

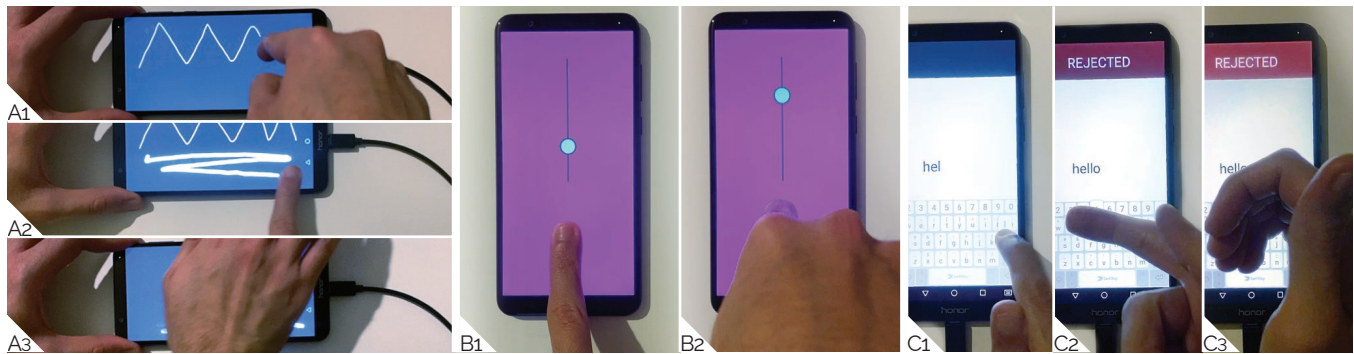


Figure 13: TouchPose can power diverse touch interactions. (A) This finger- and hand part-specific drawing app (A1–2) infers stroke width from attack angle or (A3) smudging. (B) This 3D joystick is quick to operate through finger angle. (C) Inadvertent touch rejection allows (C1) regular typing but rejects (C2) fingers at wrong orientations or (C3) other hand parts.

(Fig. 12). We hypothesize that higher-resolution touchscreens in the future and improved touch-sensing ranges [26] will provide more 3D cues on the shape of the hands. This will help alleviate single-touch ambiguities and resolve hovering fingers.

In the future, we hope to add temporal consistency to our model and correct for minor discrepancies between the touch points on the screen and the estimated fingertips positions. Incorporating the inverse kinematic [83] directly into the training procedure can help with that. Finally, rather than discriminative deep neural networks, the efficacy of generative methods can also be explored to provide multiple plausible solutions. These could be evaluated through a qualitative user study on their plausibility.

Lastly, TouchPose is enabled by a multi-task convolutional neural network-based architecture. While it could directly run on smartphones (e.g., using Tensorflow Lite), it produces computational overhead compared to standard touch event detection pipelines that are designed for low latency. We expect that future improvements in smartphone processors (especially neural inference engines) will allow TouchPose to be integrated into firmware and provide a seamless interactive experience.

9 CONCLUSION

We have presented the first general-purpose estimator for hand pose recovery using capacitive images from a touch surface alone. Our neural network TouchPose performs this task by implementing a multi-task architecture, predicting depth maps, hand poses and validity of touch events. For data acquisition, we devised a capture rig that integrates a mutual-capacitance touch sensor, a short-range depth camera below, and a stereo IR camera above for hand pose labeling. 10 participants provided touch events using various hand poses and combinations of fingers, touching, sliding, and gesturing on the touch surface while our apparatus captured synchronized frames from all sensors. Our final dataset comprises 65,374 pairs of capacitive images, depth maps and annotated 3D hand poses and served for training and testing our TouchPose model. TouchPose estimates 3D hand poses from just the intensities and imprints left on the surface, reaching an average end point error of 21.8 mm across the hand joints, including those that lie outside the touch area. In addition, TouchPose generalizes to hand gestures it has never seen

before as we demonstrated in this paper. Taken together, we believe that TouchPose will help future developers to effortlessly attach interactive behavior to input *semantics*, advancing from today’s naïve notion of touch input as 2D coordinates and interpreting input in the context of dexterous hand pose.

REFERENCES

- [1] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. 2021. Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [2] Azure Kinect 2019. Azure Kinect DK hardware specifications. <https://docs.microsoft.com/en-us/azure/kinect-dk/hardware-specification>
- [3] Gary Barrett and Ryomei Omote. 2010. Projected-capacitive touch technology. *Information Display* 26, 3 (2010), 16–21.
- [4] Alan Bränzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Lühne, René Meusel, Stephan Richter, and Patrick Baudisch. 2013. GravitySpace: Tracking Users and Their Poses in a Smart Room Using a Pressure-Sensing Floor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 725–734. <https://doi.org/10.1145/2470654.2470757>
- [5] Leslie Casas, Nassir Navab, and Stefanie Demirci. 2019. Patient 3D body pose estimation from pressure imaging. *International journal of computer assisted radiology and surgery* 14, 3 (2019), 517–524.
- [6] Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. CapStones and ZebraWidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2189–2192.
- [7] Arunesh Chandra, Pankaj Chandra, and Surinder Deswal. 2011. Hand anthropometric survey of male industrial workers of Haryana state (India). *International journal of industrial and systems engineering* 9, 1 (2011), 98–120.
- [8] Theocharis Chatzis, Andreas Stergioulas, Dimitrios Konstantinidis, Kosmas Dimitropoulos, and Petros Daras. 2020. A Comprehensive Study on Deep Learning-Based 3D Hand Pose Estimation Methods. *Applied Sciences* 10, 19 (2020), 6850.
- [9] 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.
- [10] Se-Joon Chung, Junggon Kim, Shangchen Han, and Nancy S Pollard. 2015. Quadratic Encoding for Hand Pose Reconstruction from Multi-Touch Input. In *Eurographics (Short Papers)*. 13–16.
- [11] Henry M Clever, Zackory Erickson, Ariel Kapusta, Greg Turk, Karen Liu, and Charles C Kemp. 2020. Bodies at Rest: 3D Human Pose and Shape Estimation from a Pressure Image using Synthetic Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6215–6224.
- [12] 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*. 161–166.
- [13] Xiaoming Deng, Shuo Yang, Yinda Zhang, Ping Tan, Liang Chang, and Hongan Wang. 2017. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224* (2017).

- [14] Paul Dietz and Darren Leigh. 2001. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. 219–226.
- [15] Julien Epps, Serge Lichman, and Mike Wu. 2006. A study of hand shape use in tabletop gesture interaction. In *CHI'06 extended abstracts on human factors in computing systems*. 748–753.
- [16] Kentaro Fukuchi and Jun Rekimoto. 2002. Interaction techniques for smartskin. In *Adjunct Proceedings of UIST*. 49–50.
- [17] Lihao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2017. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1991–2000.
- [18] Lihao 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.
- [19] 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), 1–15.
- [20] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. 2019. Accurate and efficient 3D hand pose regression for robot hand teleoperation using a monocular RGB camera. *Expert Systems with Applications* 136 (2019), 327–337.
- [21] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. 2019. Large-scale multiview 3d hand pose dataset. *Image and Vision Computing* 81 (2019), 25–33.
- [22] Tobias Stove-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oscar Bechtold, Steve Hodges, Matthew S Reynolds, and Joshua R Smith. 2017. Finding common ground: A survey of capacitive sensing in human-computer interaction. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3293–3315.
- [23] Jiajun Gu, Zhiyong Wang, Wanli Ouyang, Jiafeng Li, Li Zhuo, et al. 2020. 3D Hand Pose Estimation with Disentangled Cross-Modal Latent Space. In *The IEEE Winter Conference on Applications of Computer Vision*. 391–400.
- [24] Seungju Han and Joonah Park. 2013. Grip-Ball: A spherical multi-touch interface for interacting with virtual worlds. In *2013 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 600–601.
- [25] Chris Harrison, Robert Xiao, Julia Schwarz, and Scott E Hudson. 2014. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2913–2916.
- [26] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-touch sensing for mobile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2869–2881.
- [27] Ken Hinckley and Mike Sinclair. 1999. Touch-sensing input devices. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 223–230.
- [28] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 581–590.
- [29] Christian Holz and Patrick Baudisch. 2013. Fiberio: a touchscreen that senses fingerprints. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 41–50.
- [30] Christian Holz, Senaka Buttipitiya, and Marius Knaust. 2015. Bodyprint: Biometric user identification on mobile devices using the capacitive touchscreen to scan body parts. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 3011–3014.
- [31] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.
- [32] Hsin-Liu Kao, Christian Holz, Asta Roseway, Andres Calvo, and Chris Schmandt. 2016. DuoSkin: rapidly prototyping on-skin user interfaces using skin-friendly materials. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. 16–23.
- [33] 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*. 167–176.
- [34] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [35] Paul G Kry and Dinesh K Pai. 2008. Grasp recognition and manipulation with the tango. In *Experimental Robotics*. Springer, 551–559.
- [36] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the palm as an additional input modality on commodity smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [37] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 779–792.
- [38] Huy Viet Le, Sven Mayer, and Niels Henze. 2019. Investigating the feasibility of finger identification on capacitive touchscreens using deep learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 637–649.
- [39] Rui Li, Zhenyu Liu, and Jianrong Tan. 2019. A survey on 3D hand pose estimation: Cameras, methods, and datasets. *Pattern Recognition* 93 (2019), 251–272.
- [40] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021*. 1471–1482.
- [41] Noah Lockwood and Karan Singh. 2012. Fingerwalking: motion editing with contact-based hand performance. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*. 43–52.
- [42] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. 2014. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International conference on image processing (ICIP)*. IEEE, 1565–1569.
- [43] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2640–2649.
- [44] Nobuyuki Matsushita and Jun Rekimoto. 1997. HoloWall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*. 209–210.
- [45] Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the finger orientation on capacitive touchscreens using convolutional neural networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 220–229.
- [46] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 593–597.
- [47] Microsoft. 2020. *Azure Kinect Sensor SDK 1.4.1*. <https://github.com/microsoft/Azure-Kinect-Sensor-SDK>
- [48] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Generated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.
- [49] 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 the IEEE International Conference on Computer Vision Workshops*. 1284–1293.
- [50] Sara Mulatto, Alessandro Formaglio, Monica Malvezzi, and Domenico Prattichizzo. 2012. Using postural synergies to animate a low-dimensional hand avatar in haptic simulation. *IEEE transactions on haptics* 6, 1 (2012), 106–116.
- [51] Markus Oberweger and Vincent Lepetit. 2017. Deeprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 585–594.
- [52] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In *Bmvc*, Vol. 1. 3.
- [53] Keunwoo Park, Sunbum Kim, Youngwoo Yoon, Tae-Kyun Kim, and Geehyuk Lee. 2020. DeepFisheye: Near-Surface Multi-Finger Tracking Technology Using Fisheye Camera. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1132–1146.
- [54] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 113–120.
- [55] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2575–2584.
- [56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
- [57] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).
- [58] Stefano Scheggi, Leonardo Meli, Claudio Pacchierotti, and Domenico Prattichizzo. 2015. Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering. In *ACM SIGGRAPH 2015 Posters*. 1–1.
- [59] Johannes Schöning, Peter Brandl, Florian Daiber, Florian Ehtler, Otmar Hilliges, Jonathan Hook, Markus Löchtefeld, Nima Motamed, Laurence Muller, Patrick Olivier, et al. 2008. Multi-touch surfaces: A technical guide. *Johannes Schöning, Institute for Geoinformatics University of Münster, Technical Report TUM-10833* (2008).
- [60] Julia Schwarz, Robert Xiao, Jennifer Mankoff, Scott E Hudson, and Chris Harrison. 2014. Probabilistic palm rejection using spatiotemporal touch features and iterative classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009–2012.
- [61] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al.

2015. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 3633–3642.
- [62] Don A Speck, Gareth J McCaughan, and Bob L Mackey. 2007. Capacitive sensing pattern. US Patent 7,202,859.
- [63] 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.
- [64] Paul Strelt and Christian Holz. 2021. CapContact: Super-Resolution Contact Areas from Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, Article 289. <https://doi.org/10.1145/3411764.3445621>
- [65] GyberGlove Systems. 2019. CyberGlove III. <http://www.cyberglovesystems.com/>.
- [66] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. 2015. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE international conference on computer vision*. 3325–3333.
- [67] UltraLeap. 2020. Leap Motion Controller optical hand tracking module. <https://www.ultraleap.com/product/leap-motion-controller/>
- [68] Vicon. 2020. Vicon. Retrieved 2020 from <https://vicon.com/>
- [69] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui-Shyong Yeo, William Field, and Haiyan Zhang. 2018. Project zanzibar: A portable and flexible tangible interaction platform. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [70] Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao, et al. 2009. Mouse 2.0: multi-touch meets the mouse. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 33–42.
- [71] Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Øvergård, and Jan Borchers. 2015. PERCs: persistently trackable tangibles on capacitive multi-touch displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 351–356.
- [72] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. 2009. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 23–32.
- [73] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the accuracy and robustness of the leap motion controller. *Sensors* 13, 5 (2013), 6380–6393.
- [74] 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*. 2991–3000.
- [75] Andrew D Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. 2008. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 67–76.
- [76] Raphael Wimmer. 2010. Grasp sensing for human-computer interaction. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 221–228.
- [77] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.
- [78] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M Kitani. 2020. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1147–1160.
- [79] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3d finger angle on commodity touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. 47–50.
- [80] Chi Xu, Lakshmi Narasimhan Govindarajan, Yu Zhang, and Li Cheng. 2017. Lie-X: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision* 123, 3 (2017), 454–478.
- [81] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Lihao Ge, et al. 2018. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2636–2645.
- [82] Yang Zhang, Michel Pahud, Christian Holz, Haijun Xia, Gierad Laput, Michael McGuffin, Xiao Tu, Andrew Mittereder, Fei Su, William Buxton, et al. 2019. Sensing posture-aware pen+ touch interaction on tablets. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [83] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020. Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5346–5355.