# Estimating 3D Finger Angle via Fingerprint Image

KE HE, Department of Automation, BNRist, Tsinghua University, China
YONGJIE DUAN, Department of Automation, BNRist, Tsinghua University, China
JIANJIANG FENG*, Department of Automation, BNRist, Tsinghua University, China
JIE ZHOU, Department of Automation, BNRist, Tsinghua University, China

Touchscreens are the primary input devices for smartphones and tablets. Although widely used, the output of touchscreen controllers is still limited to the two-dimensional position of the contacting finger. Finger angle (or orientation) estimation from touchscreen images has been studied for enriching touch input. However, only pitch and yaw are usually estimated and estimation error is large. One main reason is that touchscreens provide very limited information of finger. With the development of under-screen fingerprint sensing technology, fingerprint images, which contain more information of finger compared with touchscreen images, can be captured when a finger touches the screen. In this paper, we constructed a dataset with fingerprint images and the corresponding ground truth values of finger angle. We contribute with a network architecture and training strategy that harness the strong dependencies among finger angle, finger region, finger type, and fingerprint ridge orientation to produce a top-performing model for finger angle estimation. The experimental results demonstrate the superiority of our method over previous state-of-the-art methods. The mean absolute errors of the three angles are 6.6 degrees for yaw, 7.1 degrees for pitch, and 9.1 degrees for roll, markedly smaller than previously reported errors. Extensive experiments were conducted to examine important factors including image resolution, image size, and finger type. Evaluations on a set of under-screen fingerprints were also performed to explore feasibility in real-world applications. Code and a subset of the data are publicly available.

CCS Concepts: • **Human-centered computing → Interaction techniques**; • **Computing methodologies → Multi-task learning**.

Additional Key Words and Phrases: finger angle, fingerprints, neural network

**ACM Reference Format:**
Ke He, Yongjie Duan, Jianjiang Feng, and Jie Zhou. 2022. Estimating 3D Finger Angle via Fingerprint Image. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 14 (March 2022), 22 pages. https://doi.org/10.1145/3517243

## 1 INTRODUCTION

Over the last years, touch interfaces have become the most common interaction mechanism for mobile devices. Capacitive sensing has become ubiquitous on mobile, wearable, and stationary devices and plays a prominent role in human-computer interaction research [9]. Despite of high freedom degrees of human fingers, the output of touchscreens is merely a two-dimensional touchpoint. Researchers have explored various way to enrich the input vocabulary of touchscreens, such as the manipulatory force [3, 33], finger shape [19], part of hand [10], touching

*Jianjiang Feng is the corresponding author

Authors' addresses: Ke He, Department of Automation, BNRist, Tsinghua University, Beijing, China; Yongjie Duan, Department of Automation, BNRist, Tsinghua University, Beijing, China; Jianjiang Feng, jfeng@tsinghua.edu.cn, Department of Automation, BNRist, Tsinghua University, Beijing, China; Jie Zhou, Department of Automation, BNRist, Tsinghua University, Beijing, China.


2474-9567/2022/3-ART14 $15.00
https://doi.org/10.1145/3517243

area size [4], 3D hand pose [5], and finger angle (or orientation) [14, 34]. Such additional input dimensions allow users to do more in the same small space of touchscreens, thereby enabling broader applications.

Among these new touch inputs, finger angle has gained a lot of attention from researchers. Finger angle (see Figure 1) refers to the three 3D angles of the finger while touching on the touchscreen: roll, pitch, and yaw. Roll is the angle around the finger's longitudinal axis, pitch is the angle between the finger and the horizontal touch surface, and yaw is the angle between the finger and the vertical axis. Several works have suggested that finger angle can be used to enrich touch input and provide additional degrees of freedom for touch interaction [6, 31, 37]. As shown in [34], with the assistance of finger angles, the user can conveniently manipulate 3D models on smartphone and adjust the volume and zoom the image on the limited screen of smartwatch. Finger angle can also be used to improve the accuracy of touch points [12]. In addition to objective performance, some researchers proposed that subjective qualities such as comfort could not be overlooked. Gil et al. [7] contributed a set of design recommendations for comfortable and effective smartwatch input via finger angle. Mayer et al. [17] further investigated the ergonomic constraints for using finger angle as additional input in a two-handed smartphone scenario. The study of [8] shows the differences in orientation and range for different fingers, hands, and actions.

Recent researches described various algorithms for estimating finger angle such that it might be used as an additional input modality for mobile applications (summarized in Table 1). Roudaut et al. [25] presented the MicroRolls algorithm, which was characterized by zero tangential velocity of the skin relative to the screen surface, to recognize at least 16 elemental gestures. They also used the roll of the thumbs as additional touch-screen input vocabulary. Watanabe et al. [33] used a camera fixed on the fingertip to monitor the light intensity emitted from the fingernail which served as a cue to computing the pitch and yaw angles. Kratz et al. [14] employed a depth camera oriented towards the device's touchscreen to generate point clouds of a finger, which were then fitted onto a cylindrical model, and eventually, yaw and pitch angles were estimated. These methods require additional sensors to obtain finger posture information, which is the primary barrier to their practical implementation. Xiao et al. [34] proposed an algorithm that utilizes 42 features extracted from capacitive images to estimate pitch and yaw. Mayer et al. [16] used a Convolutional Neural Network (CNN) with modern L2 regularization and batch normalization to improve the expressiveness of their models and showed an improvement of yaw and pitch estimation precision than empirical methods. Despite recent advances, finger angle estimation without the use of supplementary sensors remains an open problem. Existing approaches use low-resolution, information-poor capacitive images as input, which limits the accuracy of angle estimation. Additionally, to the best of our knowledge, there is no former study that provides quantitative estimation for the finger roll angle. It is worth mentioning that roll angle estimation is valuable not only because it is a part of the full 3D finger pose, but also because rolling fingers is more user-friendly and ergonomic for long-time usage compared with twisting the wrist to perform yaw.

Thanks to the rapid progress of under-screen (or in-display) fingerprint sensing technology, which seamlessly combines display and fingerprint sensing, high-resolution fingerprints are now available when a finger interacts with the touchscreen. Peng et al. [20] developed a 40 MHz ultrasonic fingerprint sensor to capture images of fingerprint and finger vessel with a resolution of 500 ppi. Yin et al. [35] developed a $368 \times 184$ optical underdisplay fingerprint sensor using the 0.11-$\mu$m CIS technology. In the commercial realm, under-screen fingerprint technology is already adopted by smartphones of many brands[1]. Qualcomm has released an ultrasonic in-display fingerprint sensor with a sensing area of 600 $mm^2$ in 2019[2]. Vivo launched a concept smartphone in 2019, which was featured with full-display fingerprint scanning technology to enable fast unlock from the entire screen[3]. While the finger

[1]https://www.androidauthority.com/phones-with-in-display-fingerprint-scanner-915950/
[2]https://www.qualcomm.com/products/3d-sonic-sensor-max
[3]https://www.pocket-lint.com/phones/reviews/vivo/147402-vivo-apex-2019-review-concept-phone

is pressed on the screen with different 3D angles, the resulting fingerprint image will change correspondingly. Unlike low-resolution capacitive images from touchscreens, ridge patterns and fine outline of the fingerprint communicate lots of information of finger angle, making fingerprint sensor a very promising modality for finger angle estimation.

In this paper, we explore the feasibility of estimating 3D finger angle via fingerprint images which provides a new perspective on human-computer interaction. A dataset consisting of 54,285 high-resolution fingerprint images and corresponding ground truth values of finger angle were collected. An approach for finger angle estimation using deep neural network was proposed and multi-task learning (MTL) was adopted to boost the performance of the network. We made use of the high interdependencies between finger angle, finger region, finger type, and fingerprint ridge orientation to minimize the complexity of the proposed network and to give informative priors to the network. State-of-the-art methods, including Gaussian Regression [34], deep neural networks [16], etc., were re-implemented and adapted to our dataset. The experimental results showed that our method outperformed all baselines and archived lowest errors for all three angles. We also conducted comprehensive experiments to investigate the contribution of individual components in our proposed network and to analyze important factors including resolution and size of fingerprint images, and different fingers. Several rules derived from these studies have been summarized. In addition we evaluated our method on a set of fingerprints collected by under-screen fingerprint sensor. Code and a subset of the data are made publicly available to facilitate further research[4].

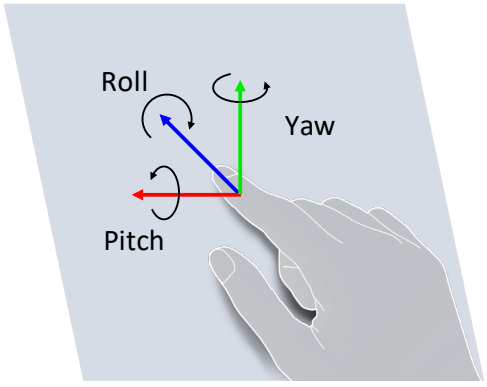



Fig. 1. The definition of finger angle: yaw, pitch, and roll.

## 2 RELATED WORKS

As previously stated, there are a variety of approaches for estimating finger angle. We divide these approaches into three categories based on the input modality: touchscreen, additional camera, and fingerprint sensor. An overview of existing researches is given in Table 1.

### 2.1 Touchscreen

Capacitive images reflected the disturbances in a projected electric field caused by finger touching, which is the most common modality in finger angle assessment since it can be obtained directly from commercial touchscreens. Wang et al. [32] estimated the yaw angle of a finger based on the shape of the capacitive image. Rogers et al. [23] further considered the pitch and presented a finger-tracking system for touch-based interaction which can track

[4]code: https://github.com/hekq/3DFingerPose, dataset: http://ivg.au.tsinghua.edu.cn/data.php

Table 1. A review of finger angle estimation researches. The dash symbol in *Ground Truth Finger Angle* means that there is no objective and accurate ground truth. The dash symbol in the *Errors* column means quantitative performance is not reported.

| Study | Year | Sensor | Ground Truth Finger Angle | Algorithm | Angles | | | Errors | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Yaw | Pitch | Roll | Yaw | Pitch | Roll |
| Wang et al. [32] | 2009 | multi-touch tabletop | - | shape-based | Y | | | - | - | - |
| Holz and Baudisch [12] | 2010 | multi-touchpad | - | similarity search | Y | | | - | - | - |
| Dang and André [6] | 2011 | multi-touch tabletop | - | shape-based | Y | | | - | - | - |
| Rogers et al. [23] | 2011 | capacitive sensor array | - | particle filter | Y | | | - | - | - |
| Zaliva [37] | 2012 | touchscreen | - | shape-based | Y | Y | Y | - | - | - |
| Watanabe et al. [33] | 2012 | additional camera | - | light intensity detection | Y | Y | Y | - | - | - |
| Kratz et al. [14] | 2013 | additional depth camera | - | point cloud registration | Y | Y | | - | - | - |
| Xiao et al. [34] | 2015 | touchscreen | wedges | feature-based | Y | Y | | 26.8 | 9.7 | - |
| Mayer et al. [18] | 2017 | additional depth camera | three RGB cameras | point cloud registration | Y | Y | | 12.2 | 15.4 | - |
| Mayer et al. [16] | 2017 | touchscreen | optical tracker | CNN | Y | Y | | 18.3 | 9.9 | - |
| Ours | - | optical fingerprint scanner | optical tracker | Multi-Task CNN | Y | Y | Y | 6.6 | 7.1 | 9.1 |

3D finger angle in addition to two-dimensional position. Zaliva [37] defined a set of useful features including area, average intensity, centroids, and the asymmetry of the shape to compute the three finger angles. The predicted finger angle was used for gesture recognition, but the angle estimation themselves were not evaluated. Xiao et al. [34] described the capacitive images as a "comet" shape. Based on this, 42 features including the ellipsoid's direction and magnitude were extracted and fed into a Gaussian Regression Model for angle estimation. Mayer et al. [16] trained convolutional neural networks for angle estimation supervised by ground truth recorded with a high-precision motion capture system. Benefitting from the powerful feature representing ability of deep neural networks, the accuracy of finger angle estimation is further improved. From a methodological standpoint, these works shared one thing in common: they all utilized the shape of the capacitive blob images as a feature to estimate finger angle, which motivated us to use the finger region of fingerprint images as a secondary task in our multi-task training scheme. Due to the low resolution of capacitive images, these approaches have limitations in terms of performance. Furthermore, the capacitive images are insensitive to the changes of the roll orientation. Therefore, the roll prediction was not implemented in the aforementioned works. However, roll angle is very useful for interaction since it covers large angle range, nearly from -90 to 90 degrees, and is easy to input for most fingers. Angle estimation technology supporting all three angles will enable more convenient interaction in many applications such as 3D model manipulation and VR/AR.

## 2.2 Additional Camera

In addition to touchscreen capacitive images, other modalities for finger angle estimation were examined by some researchers. Kratz et al. [14] employed a depth camera to capture point clouds of a finger, which were then fitted onto a cylindrical model, and eventually, yaw and pitch angles were predicted. Mayer et al. [18] developed a working prototype with a depth camera mounted on a tablet and evaluated the accuracy of their PointPose algorithm to estimate yaw and pitch angle. Dang and André [6] used the infrared images captured by a camera sensor inside the tabletop as input and processed the contour of the contacting area to estimate the finger angle. Watanabe et al. [33] used a camera fixed on the fingertip to monitor the light intensity emitted from the fingernail which served as a cue to computing the pitch and yaw angles. However, the fact that additional data-collecting devices were required may constrain their application scenarios.

## 2.3 Fingerprint Sensor

To the best of our knowledge, the study of Holz and Baudisch [12] is the only one to introduce fingerprint-based finger pose estimation into the area of human-computer interaction. A simple heuristic is exploited that two

fingerprints of a same finger are likely to exhibit similar features if and only if the finger angles corresponding to these two images are similar when touching the devices. Based on this, they developed the RidgePad algorithm. They first compared the input fingerprint with all enrolled fingerprints of the same finger using the generic image matching algorithm SURF by Bay et al. [1]. Then the algorithm looked up the $k$ closest matches in the user's pre-enrolled profile (including fingerprints with known angles covering the whole ranges of finger angles) and the angles of these closest matches were merged to generate predicted finger angle of the input fingerprint. It is worth mentioning that they used this algorithm to predict the offset of touching point, but finger angle estimation is available as a by-product. Therefore, we re-implemented this method and treated this as a baseline. However, this method requires a profile database for each new finger, which means it can only deal with enrolled fingers. A new user has to rely on a specialized optical tracking device to record precise ground truth angles when the user presses finger on fingerprint sensor at various angles, which is not feasible for real applications. A possible solution is to utilize some clear guidance on the screen and trust in the user to visually match the angle. However, using this procedure to enroll roll angles and pitch angles is not as trivial as the yaw. On the contrary, our proposed method can estimate the finger angle of new users without any enrollment or equipment for obtaining ground truth angles.

In the field of fingerprint recognition, 2D fingerprint pose estimation has been studied. Researchers have investigated how the estimated fingerprint pose can be used in for speeding up matching algorithm in large fingerprint databases [27, 36]. However, fingerprints are treated as 2D objects in their researches, and only yaw angle is estimated.

## 3 DATA ACQUISITION

Since there is no public dataset for developing and evaluating fingerprint based 3D finger pose estimation, a new database containing a large number of fingerprint images with corresponding 3D finger angles as ground truth is required. Our acquisition system is shown in Figure 2. An optical fingerprint scanner, DF500 from Dotutech[5], is utilized to capture sequential fingerprint images, continuous fingerprint video in other words, and an optical tracking system, PST-Iris from PS-Tech[6], is employed to record the ground truth 3D angles of finger. Note that the optical tracking system is used for creating training database while only fingerprint image is required in inference phase to estimate finger angles.

### 3.1 Participants

We invited 22 volunteers (10 female), to take part in the data acquisition procedure. The age of participants ranged from 20 to 58 years old. The participants were from different occupations, including manual labor, office worker, and college student, with significantly different skin conditions.

### 3.2 Procedure

For data acquisition, each participant was advised to rotate their 6 frequently used fingers, including thumb, index, and middle fingers from both left and right hands, on the fingerprint scanner respectively. First, a reflective rigid body with optical makers was attached to the participant's finger comfortably. Five additional markers were affixed on the fingerprint scanner (see Figure 2), such that the relative pose of finger to the scanner can be calculated. Participants were then instructed to use one finger to touch the scanner for each time, while changing the pitch, yaw, and roll angle of the finger slowly and steadily. A data acquisition software was developed by ourselves to collect continuous fingerprint images with corresponding 3D finger angles synchronously. Therefore a sequence of images with corresponding finger pose ground truth can be collected for each finger.

[5]http://www.dotutech.com/en/pro_d.php?id=3
[6]https://www.ps-tech.com/products-pst-iris

The participants were requested to rotate their fingers until most angles were covered by displaying the distribution of each angle during the acquisition procedure for each finger. We limited the processing operations throughout the acquisition phase to achieve a sample rate of 20 Hz. The entire fingerprint sequence (video) was saved in the buffer and written to disk after all the data has been collected for each finger. In general, it took about 2 minutes to collect fingerprint images and 3D finger angles for each finger, namely about 12 minutes per person.

### 3.3 Dataset

Finally, totally 132 (=22 × 6) fingerprint sequences with 54,285 images and corresponding ground truth values of finger angle were collected. The size of each captured fingerprint image is 800 × 750 pixels and the resolution is 500 ppi. Some collected samples together with ground truth 3D angles are shown in Figure 3. The distribution of all three angles are shown in Figure 4. Roll angles vary from −89.9° to 89.8°, pitch angle from 0.1° to 89.1°, and yaw angle from −89.5° to 88.4° (see Table 2).

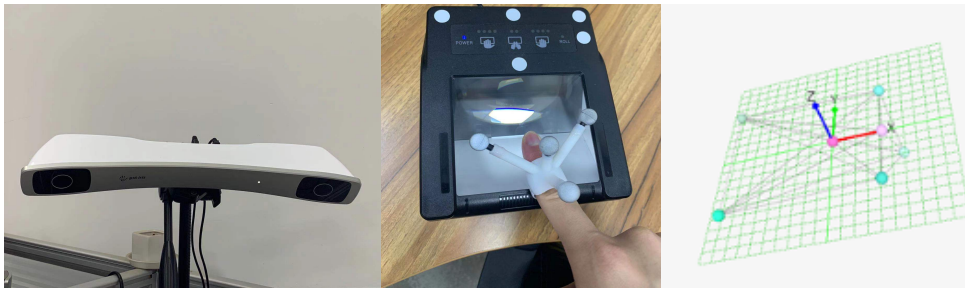


Fig. 2. The capturing system used for collecting our dataset. The left image is the PST-Iris optical tracker. The middle image shows the fingerprint scanner and a pressing finger with the reflective rigid body markers attached. The right image is a close-up of the software which shows the position of the markers in the coordinate of the scanner.

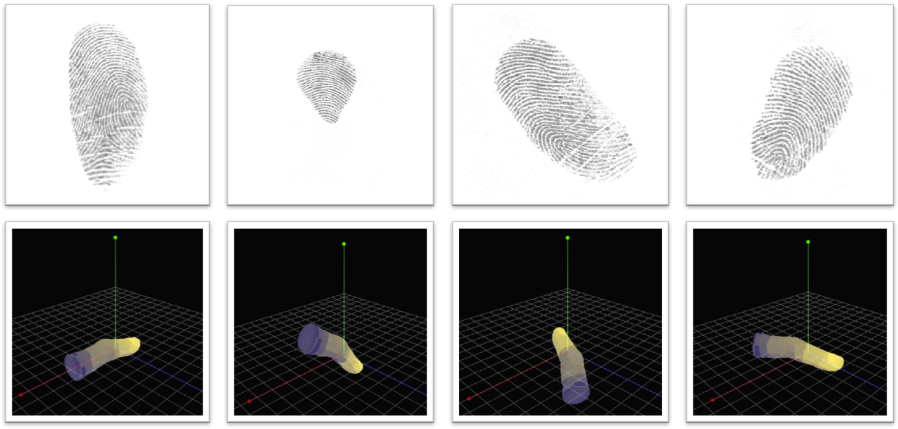
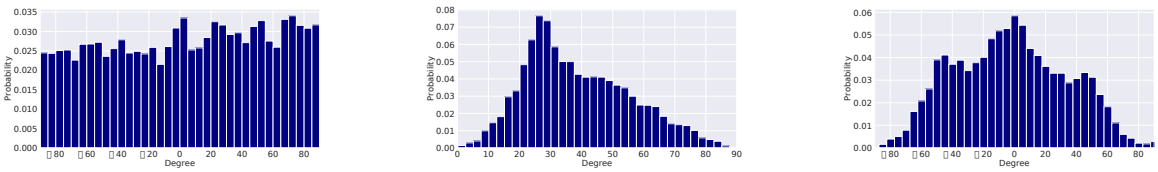


Fig. 3. Examples from the dataset. The top row shows four sample fingerprint images, and the bottom row represents the corresponding ground truth finger angle rendered in a 3D scene.




(a) yaw distribution (b) pitch distribution (c) roll distribution

Fig. 4. The distributions of three angles in our dataset.

Table 2. The basic statistics of the dataset in the experiments.

| | **Yaw** | **Pitch** | **Roll** |
|---|---|---|---|
| Min | -89.52 | 0.01 | -89.86 |
| Max | 88.35 | 89.10 | 89.79 |
| Mean | 1.61 | 35.03 | -2.02 |
| Std | 35.94 | 16.22 | 30.55 |

## 4 METHODS

In this section, we first describe the proposed multi-task CNN architecture that estimates finger angle, finger type, finger region, and fingerprint ridge orientation together (see Figure 5). Then, the angle prediction module by integrating binned angle classification and regression components is demonstrated. Finally, we illustrate the multi-loss training scheme which empowers the network to learn more meaningful feature embedding.

### 4.1 Multi-task Architecture

Multi-task learning (MTL) takes advantage of shared information from related tasks to build neural networks with better generalization ability. It has shown promising results w.r.t. performance, computations and/or memory footprint and often leads to better convergence [29].

Before the introduction of model architecture, we would like to illustrate the strategy of choosing secondary tasks. It is a simple heuristic that people, who are not familiar with fingerprints, estimate finger angle mostly based on the shape of contour given a fingerprint image. Previous works [14, 34] also utilize this as a key feature in their algorithms. As a result, the network can perform better if it is trained with the assistance of finger region segmentation task. On the other hand, the fingerprint images of thumbs, index fingers, and middle fingers are different in terms of shape, area and ridge patterns while pressed with different angles. The variation of ridge orientation is significant when rotating the fingers which can serve as an important clue to infer the finger angles. These observations motivate us to harness the strong dependencies among finger angle, finger region, finger type, and fingerprint ridge orientation.

We develop an architecture based on U-Net [24] encoder-decoder and replace its convolutional layers with bottleneck residual blocks [11]. The use of bottlenecks reduces the number of parameters and matrix multiplications. It also enables us to build a deeper network while alleviating the irritating gradient vanishing problem.

The encoder part is a ResNet backbone [11] which consists of 4 layers with 2 bottlenecks per layer. It reduces the spatial dimension of input fingerprint images from $224 \times 224$ to $1 \times 1$ pixels. In the channel dimension, we increase the number of kernels from 64 (after the first convolution) to 2048 (in the final bottleneck block). We

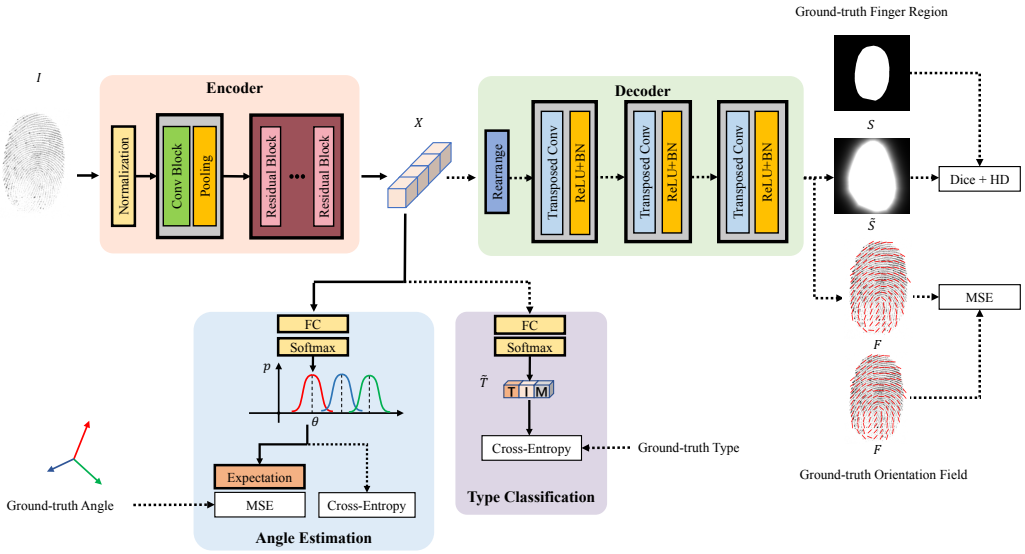



Fig. 5. Multi-task encoder-decoder network for finger angle estimation. The dotted line denotes procedures needed only in the training phase.

expect the encoder to serve as a feature embedding module that extracts a holistic fingerprint representation $X$ from the input image, allowing information to be shared across three tasks. The representation $X$ is defined as

$$X = \mathcal{F}(I|\theta) \tag{1}$$

where $\theta$ denotes the parameters of the ResNet backbone and $I$ denotes the input fingerprint image. Then a fully connected (FC) layer with parameter $\phi$ predicts the probabilities that current fingerprint image belongs to three finger types based on the feature map $X$ as follows:

$$\tilde{T} = \mathcal{C}(X|\phi). \tag{2}$$

The finger angle estimation is also computed from feature map $X$. Since it is the most important module to obtain fine-grained predictions, details will be introduced in Section 4.2.

The decoder component is composed by a re-arrange operation which is formulated as $(b, \frac{c}{64} \times 64, h, w) \rightarrow (b, \frac{c}{64}, h \times 8, w \times 8)$, and 4 layers of transposed convolution for upsampling. The final spatial resolution of the decoder subnet is $\frac{1}{4}$ of the input size. The outputs of the decoder are supervised by finger region and fingerprint ridge orientation. The decoder sub-network is as follows:

$$\{\tilde{S}, \tilde{F}\} = \mathcal{D}(X|\gamma) \tag{3}$$

where $\tilde{S}$ represents the prediction of finger region segmentation and the $\tilde{F}$ represents the estimation of fingerprint ridge orientation.

It is worth emphasizing that the location of three tasks in the whole network is essential for our multi-task architecture since the feature maps in different layers represent different levels of information in CNN. The finger angle and finger type are two global properties, whereas the finger region segmentation task and the orientation

estimation task require finer local features. Therefore, the angle estimation and type classification are executed at the end of the encoder.

It is worth emphasizing that finger type sub-module, region segmentation sub-module, and orientation estimation sub-module are only needed in the training process to assist the optimization of the neural network. In the test phase, our method requires only the encoder network and the angle prediction module to complete the finger angle estimation task (see the solid lines in Figure 5). This reduces the time cost and memory footprint.

## 4.2 Angle Prediction Module

Inspired by Hopenet [26], we propose a finger angle estimation network which takes the feature map X as input and outputs the probabilities that current fingerprint image belongs to various angle bins. The final predicts of three angles are computed in forms of expectation. We divide the domain of yaw, pitch and, roll angles into $N_1$, $N_2$, $N_3$ bins, respectively. For each Euler angle, a fully-connected (FC) layer predicts the probabilities that the current fingerprint locates at different bins,

$$\vec{p} = \mathcal{P}(X|\eta) \tag{4}$$

where $\eta$ denotes the parameters of each FC layer. To predict each angle in a fine-grained style instead of a set of discretized values, the expectation of the distribution $\vec{p}$ is computed as:

$$\tilde{y} = \sum_{k=1}^{K} p_k \cdot \mu_k \tag{5}$$

where $K$ is the number of bins, $p_k$ is the probability value for the $k$-th bin, $\mu_k$ is the representative value which is selected to be the mid-point of the interval in this paper, and $\tilde{y}$ is the final predict for each angle.

## 4.3 Multi-loss Training Scheme

We illustrate the losses for these tasks in this section.

*Angle Estimation Task.* For each Euler angle, the loss function is a combination of two components: a cross-entropy loss with label smoothing technique [21] for binned classification and a means-squared error loss function for regression. The formulation is as follows:

$$L_{\text{roll / pitch / yaw}} = L_{\text{CE}}(\vec{p}, c) + \alpha L_{\text{MSE}}(\tilde{y}, y) \tag{6}$$

where $\vec{p}$ represents the probability distribution predicted by our network, $c$ denotes the ground truth bin that the current angle belongs to, $\tilde{y}$ is the estimated angle value, and $y$ is the ground truth of the angle recorded by the optical tracking system. Each loss plays an important role in the angle estimation task. The former ensures that each fingerprint image is classified to correct interval, and the label smoothing technique keeps some tolerance to very hard samples, whereas the latter loss function guarantees our network predicts angles in a continuous domain.

*Finger Type Classification Task.* In this task, each fingerprint image is tagged with their corresponding finger type, which includes thumb, middle finger, and index finger. We use a classic cross-entropy loss function to optimize this task,

$$L_{\text{type}} = L_{\text{CE}}(\tilde{T}, t) \tag{7}$$

where $\tilde{T}$ is the aforementioned output of finger type sub-network in Equation (2) and $t$ is the ground truth finger type which is recorded in the database during collection.

*Finger Region Segmentation Task.* The ground truth segmentation of fingerprint is obtained using a traditional fingerprint region segmentation algorithm [2]. To extract high-quality foreground masks and provide precise signals which are backpropagated into the feature representation backbone, we use a combination of generalized dice loss function [28] and Hausdorff loss mentioned in [13]. The integrated loss computation for the finger region segmentation task is the following:

$$L_{\text{Haus}}(\tilde{S}, S) = \frac{1}{|\Omega|} \sum_{\Omega} \left( (\tilde{S} - S)^2 \circ \left( d_{\tilde{S}} + d_S \right) \right) \tag{8}$$

$$L_{\text{Dice}}(\tilde{S}, S) = 1 - 2 \times \frac{\sum \tilde{S} \times S + \epsilon}{\sum \tilde{S} + \sum S + \epsilon} \tag{9}$$

$$L_{\text{seg}}(\tilde{S}, S) = L_{\text{Dice}}(\tilde{S}, S) + L_{\text{Haus}}(\tilde{S}, S) \tag{10}$$

where $\tilde{S}$ is the first output of decoder Equation (3), $S$ is the ground truth finger region, $d_s$ is the Euclidean distance transformation given a finger region mask, $\Omega$ represents the area for computing Hausdorff loss which by default is the whole image, and $\epsilon$ is used to ensure the loss function stability by avoiding the numerical issue of dividing by 0.

*Fingerprint Ridge Orientation Estimation Task.* We firstly compute the ground truth fingerprint ridge orientation field of each image following the method in [22]. Then mean squared error loss function is computed between the prediction orientation field $\tilde{F}$ and the ground truth orientation field $F$ as follows:

$$L_{\text{ori}}(\tilde{F}, F) = L_{\text{MSE}}(\tilde{F}, F) = \frac{1}{|\Omega|} \sum_{\Omega} (\tilde{F} - F)^2 \tag{11}$$

where $\tilde{F}$ is the second output of decoder Equation (3), $F$ is the ground truth ridge orientation field. $\Omega$ represents the area for computing orientation field loss which by default is the fingerprint foreground region.

*Multi-Task Loss.* We design the final loss function to integrate four tasks. It is formulated as:

$$L = \alpha_{\text{angle}}(L_{\text{roll}} + L_{\text{pitch}} + L_{\text{yaw}}) + \alpha_{\text{type}} L_{\text{type}} + \alpha_{\text{seg}} L_{\text{seg}} + \alpha_{\text{ori}} L_{\text{ori}} \tag{12}$$

where the weights $\alpha_{\text{angle}}$, $\alpha_{\text{type}}$, $\alpha_{\text{seg}}$, and $\alpha_{\text{ori}}$ are used to balance the importance of four tasks. Eventually, we carry out a back-propagation, check the gradient magnitudes of each loss function about the last shared layer weights in the decoder, and refine the weights accordingly.

## 5 EXPERIMENTS

To evaluate our multi-task method, we conducted extensive experiments using the gathered fingerprint dataset. In this section, we introduce several baseline methods and their re-implementation details since some of them cannot be applied directly to our dataset. Secondly, the implementation details of our algorithm are illustrated. Then we compare the proposed method with the aforementioned baselines and analyze the results of our method. In addition, we describe the ablation studies to analyze our multi-task training scheme and report the experimental results using under-screen fingerprint sensor.

### 5.1 Experiment Settings

As mentioned in Section 3, the collected fingerprint sequences are randomly split into three subsets, 80 as training set, 26 as validation set and 26 as test set, respectively. The detailed statistics are summarized in Table 2. To inspect the generalization ability of different algorithms, we make sure that images from the same sequences are

never allowed to be dispersed in more than one subset. We want to examine whether our algorithm is capable of estimating the angle of any new fingers. For our method and other re-implemented algorithms in this paper, training dataset was used for training and the optimal models and hyper-parameters were selected based on the performance on validation dataset. Finally, all algorithms were evaluated on test dataset to make a fair comparison. Data augmentation, including random translation and rotation are applied in our method and other learning based baselines. To emulate typical under-screen fingerprints in mobile devices, we crop and downsample all the images to 224×224 (218 ppi) and the fingerprints are centered using the finger region mask (see Figure 6). To encourage further research while protecting privacy, we make the downsampled fingerprints at 55 ppi publicly available. As described in Section 5.5, the 55 ppi image is sufficient to obtain accurate angle estimates. As shown in Figure 10, the 55 ppi image does not contain detailed information such as minutiae. Furthermore, a small number of real under-screen fingerprint images by an ultrasonic sensor were also collected to explore the feasibility of our method in real applications.

Fig. 6. Reducing the size and resolution of input fingerprint. Left is the original image (800 × 750 pixels, 500 ppi), right is the cropped and downsampled image (224 × 224 pixels, 218 ppi)

As for evaluation metrics, we use both Mean Absolute Error (MAE) metric and Root Mean Squared Errors (RMSE) metric following the protocol in [16, 34]. The Standard Deviation (SD) is also reported.

## 5.2 Baseline Methods

Firstly, we re-implemented the approaches in [12, 16, 34] and adapted them for our dataset, since they were not designed for fingerprint images. We also adapted methods from other domains [11, 26] to our database to provide stronger baselines.

- In contrast to all other methods, Holz and Baudisch's method was essentially a way to search user databases based on the image similarities. Their method was originally designed to obtain a more precise touching point, but we pondered it to be capable of estimating finger angle also. This approach requires a set of fingerprints for every finger, which means for each fingerprint image in the test set, multiple fingerprints of the same finger with corresponding ground truth angles must be known. Because of this, for each sequence

in the test set, we randomly sampled some images to be the query collection, and several images to be the database. Following their work, we used the SURF algorithm to extract image descriptors for both query images and database images and finally merged the Top-$K$ closest matches in corresponding fingerprints for each query image to form a prediction, where the hyperparameter $K$ was set to 5. The whole algorithm can be formulated as:

$$w_x = \sum_{i \in \mathcal{F}_q, j \in \mathcal{F}_x} \mathbb{I}(d_i, d_j) \tag{13}$$

$$\hat{y_q} = \frac{\sum_{k=1}^{K} w_k \vec{y_k}}{\sum_{k=1}^{K} w_k} \tag{14}$$

where $d_i$ denotes the $i$-th SURF descriptor of the query image, $d_j$ representes the $j$-th SURF descriptor of the database image $x$, $w_x$ measures the number of matches which indicates the similarity score of the image $x$ in the database with the current query image $q$. To compute it, we have to loop over all possible matches between $x$ and $q$. The indicator function is conducted following the ratio test proposed by Lowe [15]. $\vec{y_q}$ is the final prediction for the current query image $q$ which is used for evaluation. The experimental results showed that the performance and time cost of this algorithm were highly sensitive to the counts of database images. When the number of available images in the user-specific database for searching was smaller than 10, we observed several failure cases in which the certain query image and all of the images in database shared no matching descriptors. This will result in the rejection of prediction since Equation 13 is equal to 0. As mentioned in Section 2.3, the fact that any new user should take an enrollment process restricts the application of this method.

- As for Xiao et al.'s method [34], we nearly reproduced all the procedures mentioned in their paper. However, the distance and angle between the touching point outputted by the touchscreen driver and the ellipsoid centroid were inaccessible because our fingerprint scanner does not output a touching point. Therefore, 28 features for each image were extracted and used for training the GPR (Gaussian Process Regressor) algorithm using the sklearn toolkits[7].
- To examine the capability of existing deep learning models on our proposed dataset, we also implemented a model which was inspired by the best model in Mayer et al. [16]. It is a Convolutional Neural Network sharing similar architecture as Mayer et al.'s model with L2 regularization and BatchNormalization. Since their model was originally designed to deal with low-resolution capacitive images, the model size was chosen to be relatively small. It is a bit unfair to compare the original model with our method, and therefore we further adjusted the ResNet [11] model by replacing the last fully-connected layer from 1000 channels to 3 channels (corresponding to yaw, pitch, and roll). It was supervised using the same regression loss function in [16]. This ResNet based model could be seen as the extension of Mayer et al.'s CNN architecture and serves as a stronger baseline.
- Finally, the Hopenet for head pose estimation [26] was adapted for our task by using our angle estimation module (4.2). It was trained and evaluated on our dataset as another baseline method. We divided the domain of yaw, pitch, and roll angles into 60, 30, 60 bins, respectively, which was the same with our angle prediction module.

[7]https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html

Table 3. The best results for all tested methods. Errors are reported in angular degree errors. It is worth mentioning that Holz and Baudisch [12] is special for its requirement of the enrollment process for any new users with optical tracking hardware. Note that the original systems in [34], [12], and [16] cannot output the roll angle. The results listed in this and later tables are generated from our re-implemented or re-designed counterpart methods.

| | Yaw | | | Pitch | | | Roll | | | Overall | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** |
| Methods without fingerprint enrollment | | | | | | | | | | | | |
| Xiao et al. [34]* | 38.60 | 35.04 | 16.18 | 15.23 | 11.73 | 9.71 | 31.18 | 24.99 | 18.64 | 29.96 | 23.92 | 18.05 |
| Mayer et al. [16]** | 21.85 | 14.99 | 15.91 | 13.42 | 10.43 | 8.45 | 18.92 | 13.92 | 12.81 | 18.40 | 13.11 | 12.91 |
| ResNet [11] | 15.70 | 9.91 | 12.25 | 10.09 | 7.73 | 6.54 | 14.15 | 9.65 | 10.38 | 13.52 | 9.10 | 10.01 |
| Hopenet [26] | 13.71 | 9.00 | 11.52 | 9.51 | 7.75 | 7.13 | 13.03 | 9.32 | 9.34 | 12.22 | 8.69 | 8.59 |
| Ours | **11.37** | **6.63** | **9.21** | **9.28** | **7.13** | **6.16** | **12.14** | **9.07** | **9.28** | **10.99** | **7.60** | **7.93** |
| Method with fingerprint enrollment | | | | | | | | | | | | |
| Holz and Baudisch [12]* | 11.01 | 6.52 | 8.14 | 8.58 | 7.04 | 7.74 | 14.01 | 10.65 | 12.60 | 11.41 | 8.07 | 8.08 |

* *re-implemented*

** *a similar model inspired by Mayer et al. [16]*

## 5.3 Implementation Details

We used Pytorch[8] for implementing the proposed multi-task approach. For data augmentation in training, we applied random translation, random cropping, and random rotation, the corresponding yaw angles were also modified accordingly. All the hyperparameters were as follows: the number of bins $N_1 = 60, N_2 = 30, N_3 = 60$, $\alpha = 0.2$ in Equation 6, the label smoothing coefficient in cross-entropy loss function is 0.1, $\epsilon = 1$ in Equation 9, $\alpha_{\text{angle}} = 1$, $\alpha_{\text{type}} = 5$, $\alpha_{\text{seg}} = 1$ $\alpha_{\text{ori}} = 10$ in Equation 12. We used 100 epochs to train the network with the AdamW optimizer, whose $betas = (0.9, 0.999)$, $init\_lr = 0.0001$, $weight\_decay = 0.001$. The batch size was set to be 256. The learning rate was reduced by a factor of 0.1 every 40 epochs. Dropout layer was used after the fully-connected layer of the angle estimation module to mitigate the overfitting problem with dropout rate of 0.1. The experiments were performed on a computer with an Intel Xeon E5 CPU and two Nvidia GTX 3090 GPUs. It took about 3 hours with data-parallel acceleration to train the models.

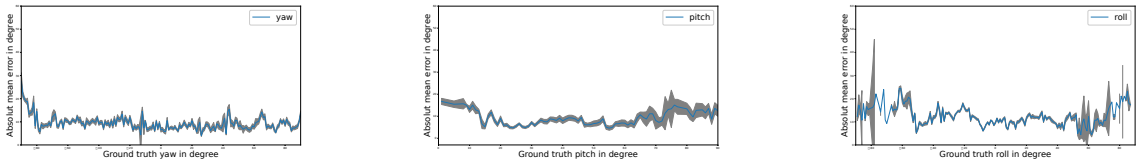
## 5.4 Comparison with Baseline Methods

In Table 3, we compared our multi-task approach with several baseline methods on the collected dataset. It is worth mentioning that Holz and Baudisch [12] is special for its requirement of the enrollment process for any new users, which means extra equipment or user guidance are required. The remaining methods don't have any prerequisites except for the fingerprint images. As shown in the table, methods driven by deep learning performs better than traditional machine learning algorithms. This makes sense since the powerful feature representation of deep neural networks. We found that Hopenet [26] outperforms ResNet [11] in this angle estimation task not only from the perspective of average error. The standard deviation of the Hopenet is also lower than ResNet. This observation shows that the combination of binned angle classification and regression objectives is more robust than direct angle regression. Our proposed multi-task encoder-decoder architecture further boosts the final performance. Additionally, the performances for three auxiliary tasks are as follows: the average classification accuracy for 3 finger types (thumb, index, middle) is 0.76, the mean estimation error for ridge orientation is 5.42 degrees per pixel, and the average dice coefficient for segmentation sub-task is 0.95.

## 5.5 Performance Analysis

In this section, we conduct a comprehensive analysis of our proposed multi-task encoder-decoder network.

[8]https://pytorch.org/




(a) yaw error distribution (b) pitch error distribution (c) roll error distribution

Fig. 7. The error distributions of three angles in our dataset. The gray area shows 95% CL.

**Error distributions:** As shown in Figure 7, we found the proposed method performed well in -40° to 40° for yaw, -60° to -20° for pitch and -50° to 50° for roll. Given the distributions of our data collection (as shown in Figure 4), this error distribution is reasonable since the performance of neural networks is highly impacted by the data distribution. Extremely large pitch and roll angles will result in very small fingerprint images, making it difficult to accurately estimate the finger angle.

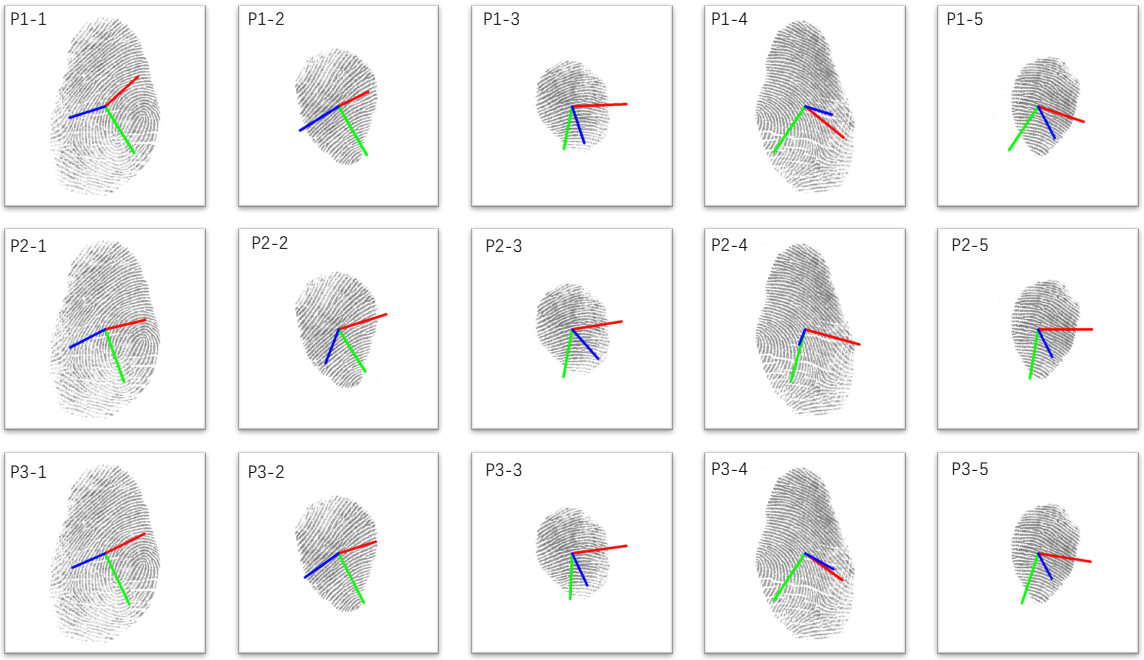



Fig. 8. Examples of finger angle estimation. From P1-1 to P1-5, they are ground truths. From P2-1 to P2-5 are the results of Mayer et al. [16]. From P3-1 to P3-5 are our results. The blue line indicates the longitudinal axis (roll), the green line indicates the vertical axis (yaw), and the red line indicates the transverse axis (pitch).

**Errors for different fingers:** Table 4 showed the errors for different fingers. As seen in the table, the performance of yaw and roll for the thumb is the best. We pondered that this was because along with the variation

Table 4. Performance of different fingers. MAE is used as the metric of evaluation.

| | Yaw | Pitch | Roll | Overall |
|---|---|---|---|---|
| thumb | **5.83** | 8.24 | **8.42** | 7.50 |
| index | 7.29 | **6.23** | 8.85 | **7.46** |
| middle | 6.77 | 6.92 | 9.94 | 7.87 |

Fig. 9. Fingerprints of different effective area sizes. From left to right, they are full-sized image with ground truth, full-sized image with prediction, 80%, 60%, 40% images with predictions.

Table 5. Performance on fingerprints of different effective area sizes. MAE is used as the metric of evaluation.

| | Yaw | Pitch | Roll | Overall |
|---|---|---|---|---|
| 100% | **6.63** | **7.13** | **9.07** | **7.60** |
| 80% | 6.93 | 7.49 | 10.28 | 8.23 |
| 60% | 8.51 | 8.16 | 13.69 | 10.12 |
| 40% | 13.93 | 10.25 | 17.39 | 13.85 |

of finger angle, the fingerprint images of thumbs changed most significantly and thus provide more clues for inferring finger angle. Therefore, in turn, it is easier for the network to estimate angle from thumb fingerprints.

**Errors for different effective area sizes:** Large under-screen fingerprint sensors are expensive and not as popular as small area sensors. To imitate smaller under-screen fingerprints and examine the performance of our approach on them, we conducted three additional experiments. First, we constructed three datasets by randomly cropping the downsampled images (218 ppi) into 40%, 60%, and 80% respectively. The resulting image samples are shown in Figure 9. We trained and evaluated our model on these datasets and the results are summarized in Table 5. The errors increase a little from full-sized fingerprints to fingerprints of 60% area. However, the proposed method performed significantly worse when the area of fingerprints was cropped to only 40%. It is not surprising since a lot of information (including finger boundary and ridge pattern) has been dropped (see the rightmost image of Figure 9).

**Errors for different resolutions:** Mobile applications are concerned with speed and memory footprint. Reducing the resolution of the image is beneficial for increasing frame rates and reducing delay in interaction. To examine the robustness of our approach to deal with images of different resolutions, we downsampled all the images from 218 ppi ($224 \times 224$ pixels) to 109 ppi ($112 \times 112$ pixels), and 55 ppi ($56 \times 56$ pixels) respectively. Our multi-task neural network were then trained and evaluated on these datasets and the quantitative results are given in Table 6. Figure 10 shows a fingerprint in different resolutions and their corresponding predictions. As shown in the Table 6, Xiao et al.'s method remains unaltered with various resolutions since their method is basically based on the shape of fingerprint contour, which is nearly unchanged for different resolutions. Mayer et al.'s method performs slightly better on higher resolution images. This is mainly because their neural network is relatively shallow and cannot learn detailed features available in images of higher resolution than 55 ppi (56

Fig. 10. Fingerprints of different resolutions. From left to right, they are 218 ppi image with ground truth, 218 ppi image with prediction, 109 ppi image with prediction, and 55 ppi image with prediction. The axes in the rightmost image were blurry due to the low resolution.

Table 6. Performance on fingerprints of different resolutions. MAE is used as the metric of evaluation.

| | 218 ppi | | | | 109 ppi | | | | 55 ppi | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **Yaw** | **Pitch** | **Roll** | **Overall** | **Yaw** | **Pitch** | **Roll** | **Overall** | **Yaw** | **Pitch** | **Roll** | **Overall** |
| Methods without fingerprint enrollment | | | | | | | | | | | | |
| Xiao et al. [34]* | 35.04 | 11.73 | 24.99 | 23.92 | 35.78 | 11.84 | 24.78 | 24.13 | 37.17 | 12.40 | 25.85 | 25.14 |
| Mayer et al. [16]** | 14.99 | 10.43 | 13.92 | 13.11 | 15.23 | 11.20 | 14.33 | 13.59 | 15.69 | 10.93 | 15.50 | 14.04 |
| Ours | **6.63** | **7.13** | **9.07** | **7.60** | **7.53** | **8.44** | **10.59** | **8.85** | **10.72** | **8.94** | **15.91** | **11.86** |
| Method with fingerprint enrollment | | | | | | | | | | | | |
| Holz and Baudisch [12]* | 6.52 | 7.04 | 10.65 | 8.07 | 8.39 | 11.69 | 19.05 | 13.04 | 10.88 | 11.17 | 24.59 | 15.55 |

* *re-implemented*
** *a similar model inspired by Mayer et al. [16]*

× 56 pixels). Our model outperforms all baselines and even exceeds the Holz and Baudisch's method which relies fingerprint enrollment in the test phase. It is probably because SURF feature matching does not perform well on very low resolution fingerprints. We also observed that the yaw and pitch angles are not sensitive to the resolution. Whereas the error of the roll angle increases significantly with the degradation of resolution. In conjunction with previous experiments, it is not hard to conclude that the roll angle is the hardest one for estimation, and it highly relies on the fine texture information of fingerprint images. Both lower resolution and smaller effective area will result in poorer performance of roll angle. This conclusion reveals the importance of the high-resolution fingerprint images for finger angle estimation task. It may also explain why roll angle is relatively neglected in previous touchscreen based finger angle estimation studies.

**Overfitting analysis:** To analyze possible over-fitting problems of the proposed model, we report the performance of the trained model on training set, validation set, and test set in Table 7 (The performance on the test set is the same as Table 3.). All of the experimental settings and conditions were held the same as the main study (see Section 5.1). As seen in Table 7, the overfitting phenomenon does exist, but is not severe. This can be attributed to multiple training skills we used, including the weight decay strategy, label smoothing technique, and dropout operation (see Section 5.3 for more details).

## 5.6 Ablation Study

To investigate the contribution of individual components in our proposed method, we conducted ablation evaluations on the dataset.

Table 7. Overfitting analysis.

| | Yaw | | | Pitch | | | Roll | | | Overall | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset split** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** | **RMSE** | **MAE** | **SD** |
| training set | 7.36 | 3.31 | 4.95 | 8.17 | 3.78 | 5.34 | 10.86 | 7.26 | 8.49 | 8.92 | 4.78 | 6.45 |
| validation set | 10.78 | 6.42 | 8.92 | 10.11 | 7.42 | 6.82 | 13.21 | 9.47 | 9.76 | 11.44 | 7.77 | 8.59 |
| test set | 11.37 | 6.63 | 9.21 | 9.28 | 7.13 | 6.16 | 12.14 | 9.07 | 9.28 | 10.99 | 7.60 | 7.93 |

Table 8. Ablation study of different regression loss coefficients. MAE is used as the metric of evaluation.

| $\alpha$ | **Yaw** | **Pitch** | **Roll** | **Overall** |
|---|---|---|---|---|
| 10 | 8.79 | 7.78 | 9.47 | 8.68 |
| 1 | 6.77 | 7.69 | **8.94** | 7.80 |
| 0.2 | **6.63** | **7.13** | 9.07 | **7.60** |
| 0.01 | 7.88 | 7.95 | 10.43 | 8.75 |
| 0.001 | 9.79 | 10.34 | 14.92 | 11.68 |

Table 9. Ablation study of the three secondary tasks. MAE is used as the metric of evaluation.

| | **Yaw** | **Pitch** | **Roll** | **Overall** |
|---|---|---|---|---|
| No Auxiliary Tasks | 9.00 | 7.75 | 9.32 | 8.69 |
| Finger Type | 8.72 | 7.13 | 9.49 | 8.45 |
| Finger Region | 6.93 | 7.69 | 9.23 | 7.95 |
| Ridge Orientation | 6.72 | 7.35 | 9.18 | 7.75 |
| Finger Type + Finger Region + Ridge Orientation | **6.63** | **7.13** | **9.07** | **7.60** |

- **Combination of binned classification and regression.** As shown in Table 3, Hopenet outperformed ResNet significantly in terms of the average error and the standard deviation. Since the main difference between these two methods is the usage of binned cross-entropy loss function, this proves the effectiveness and robustness of binned classification strategy. Even further, we trained and evaluated different models with different coefficients for the Mean Squared Error (MSE) loss component (see $\alpha$ in Equation 6) while maintaining the weight of the Cross-Entropy loss constant at 1. As shown in Table 8, the best results are achieved when the regression loss coefficient is equal to 0.2. And the errors of finger angle estimation decrease when this coefficient increases.
- **Different auxiliary tasks.** The following ablation experiments were carried out in order to thoroughly explore the respective roles of the three auxiliary tasks. As seen in Table 9, the estimation of yaw is significantly better if the finger region is segmented and ridge orientation field is estimated jointly. This makes sense because the contour of segmentation and the orientation of ridges are important clues for the estimation of the yaw angle. In fact, previous works [6, 23, 32] utilized this fact to design their angle estimation algorithm. Whereas the finger type sub-task mainly brings improvement to the pitch angle. By optimizing the three tasks together, the best performance is achieved.

## 5.7 Feasibility Analysis

To explore the feasibility of our method for real under-screen fingerprint sensors, a small number of under-screen fingerprint images were captured by an ultrasonic sensor embedded in a mobile phone. Due to limited access to this device, we cannot obtain sufficient number of images. Due to the non-disclosure agreement (NDA) with the vendor, the sensor specifications and original fingerprint images captured could not be described in detail in this paper. Similarly, optical tracking system was used to track finger pose while touching on display, which

Table 10. The basic statistics of the collected under-screen fingerprint images in the experiments.

| | Yaw | Pitch | Roll |
|---|---|---|---|
| Min | -87.80 | 1.78 | -82.85 |
| Max | 86.83 | 89.17 | 84.22 |
| Mean | 1.53 | 40.02 | -1.78 |
| Std | 39.08 | 18.34 | 46.60 |

Table 11. The best results of three methods on the under-screen fingerprint dataset. MAE is used as the metric of evaluation.

| | Yaw | Pitch | Roll |
|---|---|---|---|
| Ours | 18.7 | 9.8 | 19.0 |
| Mayer et al. [16]** | 24.3 | 17.8 | 26.2 |
| Xiao et al. [34]* | 38.0 | 21.7 | 33.5 |

* *re-implemented*

** *a similar model inspired by Mayer et al. [16]*

were used as ground truth value of finger angles. To mitigate the modality difference between under-screen fingerprint images and previous optical based fingerprint images, we utilized a histogram mapping method to match the intensity distribution of under-screen and optical fingerprint image. The intensity distributions of the original images in these two datasets reveal such modal differences (see Figure 12). An example of pre-processed under-screen fingerprint image is shown in Figure 11. We applied the proposed algorithm to evaluate the pose estimation performance on these pre-processed under-screen fingerprint images. Totally 3790 images were collected from 24 different fingers of 4 subjects, and we split them into 3 subsets for training (1989), validation (608), and test (1193), respectively. The detailed statistics are summarized in Table 10. Specifically, the network was pre-trained using optical based fingerprint images, and fine-tuned using these pre-processed under-screen fingerprint images.

The estimation errors are 18.7 degrees for yaw, 9.8 degrees for pitch, and 19.0 degrees for roll in terms of MAE. As a comparison, the baseline methods listed in Section 5.4 were also trained and evaluated on this dataset. The results are summarized in Table 11. Limited by the small number of collected under-screen fingerprint images, the very short usage time, and the modality difference between ultrasonic and optical based fingerprint images, the estimation accuracies of all methods are worse than the results in Section 5.4. We believe that this problem can be alleviated by using more under-screen fingerprint images in training and developing specific image enhancement algorithm. Because the ultrasonic sensor is not available to us, further optimization is beyond the scope of this study. Of note, the estimation error of our algorithm could be smaller in practice. As mentioned in [31], in many application scenarios only one angle or two angles are used. Even in application of 3D object manipulation, controlling one Degree of Freedom (DOF) at a time may lead to better performance than controlling all DOF simultaneously [30]. When only one or two finger angles are required, the estimation problem is easier in nature.

## 6 LIMITATION

Although the preliminary work in this paper achieved good accuracy and showed the feasibility of estimating 3D finger angles on under-screen fingerprints. The current study has the following limitations.

Our dataset has several limitations. The fingerprint images (our main dataset) were captured by a traditional optical fingerprint scanner based on frustrated total internal reflection (FTIR) technique, as opposed to under-screen fingerprint sensors used on mobile devices. It is difficult to obtain fingerprint images from smartphones through

Fig. 11. An example of under-screen fingerprint image after histogram mapping.

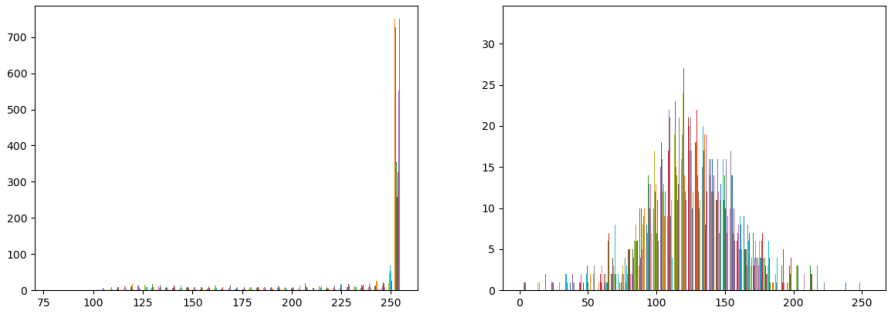



Fig. 12. The intensity distributions of original images in the optical fingerprint dataset (left) and ultrasonic fingerprint dataset (right).

software because fingerprint, as sensitive data, is processed and stored in the Trusted Execution Environment (TEE)[9]. Despite of our effort to collect 3970 under-screen images from a smartphone equipped with ultrasonic fingerprint sensor, more of them are necessary to train a reliable deep neural network. In addition, fingers that were particularly dry or wet were not intentionally collected. Including more fingerprints of low quality may lead to degradation in performance. The total sample size was also limited and larger dataset is necessary for training more generalizable model.

As shown in Figure 7, the performance of our model is unstable when the angle values are extremely large. The standard deviation of errors at extremely large angles reveal the instability for such samples. The imbalanced dataset is responsible for this. However, the disequilibrium could not be solved thoroughly since it is difficult to cover all possible combinations of the yaw, pitch, and roll angles. Therefore, it is necessary to develop more robust and efficient data augmentation strategies, which could be archived by the projection of three-dimensional finger models or data synthesis using Generative Adversarial Networks (GAN).

Besides the dataset, there is still a lot of room for refinement in our approach. For example, we did not investigate the usage of time domain smoothing (e.g., Kalman filter, particle filter, etc.) to reduce the error, which has been shown to be feasible in [23]. The inference time of our model on an Nvidia 3090 GPU is 0.0011 seconds

[9]https://source.android.com/security/trusty

per image, and the number of parameters is 27.35 M. However, the computation and memory resources are constrained on mobile devices, which means a more efficient backbone network should be explored in the future.

## 7 CONCLUSION

In this paper, we propose a multi-task deep neural network for estimating 3D finger angle via fingerprint images. To exploit the feasibility and train the model, we constructed a dataset with fingerprint images and the corresponding ground truth values of finger angle. Multi-task learning is utilized to harness the strong dependencies among finger angle, finger region, and finger type. The experimental results show that our method outperforms previous state-of-the-art methods. The errors of the three angles are 6.6 degrees for yaw, 7.1 degrees for pitch, and 9.1 degrees for roll. The proposed approach is found to be robust to fingerprint images with different resolutions and image sizes, which is important for real applications. Preliminary assessment on a small set of ultrasonic under-screen fingerprints shows promising results. With accurate and robust full 3D finger angle estimation, more innovative human-computer interactions will become possible for smartphones, tablets, and PCs.

## ACKNOWLEDGMENTS

This research was supported in part by National Key Research and Development Program of China (2018AAA0102803) and National Natural Science Foundation of China (61976121). We would like to thank all the anonymous reviewers for their constructive suggestions.

## REFERENCES

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110, 3 (June 2008), 346–359. https://doi.org/10.1016/j.cviu.2007.09.014

[2] A.M. Bazen and Sabih H. Gerez. 2001. Segmentation of Fingerprint Images. *14th ProRISC Workshop on Circuits, Systems and Signal Processing 2003* (2001), 276–280.

[3] Tobias Boceck, Sascha Sprott, Huy Viet Le, and Sven Mayer. 2019. Force Touch Detection on Capacitive Sensors Using Deep Neural Networks. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 42, 6 pages. https://doi.org/10.1145/3338286.3344389

[4] Sebastian Boring, David Ledo, Xiang'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The fat thumb: using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*. 39–48. https://doi.org/10.1145/2371574.2371582

[5] Frederick Choi, Sven Mayer, and Chris Harrison. 2021. 3D Hand Pose Estimation on Conventional Capacitive Touchscreens. *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction* (2021).

[6] Chi Tai Dang and Elisabeth André. 2011. Usage and recognition of finger orientation for multi-touch tabletop interaction. In *INTERACT'11 Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III*. 409–426.

[7] Hyunjae Gil, Hongmin Kim, and Ian Oakley. 2018. Fingers and Angles: Exploring the Comfort of Touch Input on Smartwatches. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 164 (Dec. 2018), 21 pages. https://doi.org/10.1145/3287042

[8] Alix Goguey, Géry Casiez, Daniel Vogel, and Carl Gutwin. 2018. Characterizing Finger Pitch and Roll Orientation During Atomic Touch Actions. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018).

[9] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. *Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction.* Association for Computing Machinery, New York, NY, USA, 3293–3315. https://doi.org/10.1145/3025453.3025808

[10] Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/2047196.2047279

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[12] 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 28th International Conference on Human Factors in Computing Systems - CHI '10*. ACM Press.

https://doi.org/10.1145/1753326.1753413

[13] Davood Karimi and Septimiu E. Salcudean. 2020. Reducing the Hausdorff Distance in Medical Image Segmentation With Convolutional Neural Networks. *IEEE Transactions on Medical Imaging* 39, 2 (2020), 499–513.

[14] Sven Kratz, Patrick Chiu, and Maribeth Back. 2013. PointPose: Finger Pose Estimation for Touch Input on Mobile Devices Using a Depth Sensor. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (St. Andrews, Scotland, United Kingdom) *(ITS '13)*. Association for Computing Machinery, New York, NY, USA, 223–230. https://doi.org/10.1145/2512349.2512824

[15] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (Nov. 2004), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

[16] 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*. ACM. https://doi.org/10.1145/3132272.3134130

[17] Sven Mayer, Huy Viet Le, and Niels Henze. 2018. Designing finger orientation input for mobile touchscreens. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. https://doi.org/10.1145/3229434.3229444

[18] Sven Mayer, Michael Mayer, and Niels Henze. 2017. Feasibility analysis of detecting the finger orientation with depth cameras. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM. https://doi.org/10.1145/3098279.3122125

[19] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD. Rasel Islam. 2016. *The Flat Finger: Exploring Area Touches on Smartwatches*. Association for Computing Machinery, New York, NY, USA, 4238–4249. https://doi.org/10.1145/2858036.2858179

[20] Chang Peng, Mengyue Chen, and Xiaoning Jiang. 2021. Under-Display Ultrasonic Fingerprint Recognition With Finger Vessel Imaging. *IEEE Sensors Journal* 21, 6 (March 2021), 7412–7419. https://doi.org/10.1109/jsen.2021.3051975

[21] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *ICLR (Workshop)*.

[22] Nalini K. Ratha, Shaoyun Chen, and Anil K. Jain. 1995. Adaptive flow orientation-based feature extraction in fingerprint images. *Pattern Recognition* 28, 11 (1995), 1657–1672. https://doi.org/10.1016/0031-3203(95)00039-3

[23] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2011. *AnglePose: Robust, Precise Capacitive Touch Tracking via 3d Orientation Estimation*. Association for Computing Machinery, New York, NY, USA, 2575–2584. https://doi.org/10.1145/1978942.1979318

[24] 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*. 234–241.

[25] Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. *MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb*. Association for Computing Machinery, New York, NY, USA, 927–936. https://doi.org/10.1145/1518701.1518843

[26] Nataniel Ruiz, Eunji Chong, and James M. Rehg. 2018. Fine-Grained Head Pose Estimation Without Keypoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2074–2083.

[27] Yijing Su, Jianjiang Feng, and Jie Zhou. 2016. Fingerprint indexing with pose constraint. *Pattern Recognit.* 54 (2016), 1–13. https://doi.org/10.1016/j.patcog.2016.01.006

[28] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. 2017. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. *Lecture Notes in Computer Science* (2017), 240–248. https://doi.org/10.1007/978-3-319-67558-9_28

[29] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-Task Learning for Dense Prediction Tasks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/tpami.2021.3054719

[30] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. 2009. Influence of Degrees of Freedom's Manipulation on Performances during Orientation Tasks in Virtual Reality Environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* (Kyoto, Japan) *(VRST '09)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/1643928.1643942

[31] Jonas Vogelsang, Francisco Kiss, and Sven Mayer. 2021. A Design Space for User Interface Elements Using Finger Orientation Input. In *Mensch Und Computer 2021* (Ingolstadt, Germany) *(MuC '21)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3473856.3473862

[32] 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 - UIST '09*. ACM Press. https://doi.org/10.1145/1622176.1622182

[33] Yoichi Watanabe, Yasutoshi Makino, Katsunari Sato, and Takashi Maeno. 2012. Contact force and finger angles estimation for touch panel by detecting transmitted light on fingernail. In *EuroHaptics'12 Proceedings of the 2012 International Conference on Haptics: Perception, Devices, Mobility, and Communication - Volume Part I*. 601–612.

[34] 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 - ITS '15*. ACM Press. https://doi.org/10.1145/2817721.2817737

[35] Ping-Hung Yin, Chih-Wen Lu, Jia-Shyang Wang, Keng-Li Chang, Fu-Kuo Lin, and Poki Chen. 2021. A 368 × 184 Optical Under-Display Fingerprint Sensor Comprising Hybrid Arrays of Global and Rolling Shutter Pixels With Shared Pixel-Level ADCs. *IEEE Journal of Solid-State Circuits* 56, 3 (March 2021), 763–777. https://doi.org/10.1109/jssc.2020.3042894

[36] Qihao Yin, Jianjiang Feng, Jiwen Lu, and Jie Zhou. 2021. Joint Estimation of Pose and Singular Points of Fingerprints. *IEEE Trans. Inf. Forensics Secur.* 16 (2021), 1467–1479. https://doi.org/10.1109/TIFS.2020.3036803

[37] Vadim Zaliva. 2012. 3D finger posture detection and gesture recognition on touch surfaces. In *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE. https://doi.org/10.1109/icarcv.2012.6485185