Estimating 3D Finger Pose via 2D-3D Fingerprint Matching

Yongjie Duan
BNRist, Department of Automation, Tsinghua University
China

Ke He
BNRist, Department of Automation, Tsinghua University
China

Jianjiang Feng*
BNRist, Department of Automation, Tsinghua University
China
jfeng@tsinghua.edu.cn

Jiwen Lu
BNRist, Department of Automation, Tsinghua University
China

Jie Zhou
BNRist, Department of Automation, Tsinghua University
China

ABSTRACT

Touchscreens have become the primary input devices for smartphones, tablet computers, and other intelligent devices over the past decades. While for the most pervasive commercial devices, only 2D touch positions on the screen are utilized as interaction inputs. To extend the richness of the input vocabulary, some researchers have proposed several innovative interaction techniques, e.g. finger pose. However, due to the low resolution and lacking in information of capacitive images, only two angles, pitch and yaw, are considered in most finger pose estimation algorithms, and the accuracy is not sufficiently high for large scale applications in smartphones. With the rapid development of under-screen fingerprint sensing technology, a new input modality, fingerprint image, for 3D finger pose estimation is available from these fingerprint sensors. In this paper, we propose a finger specific algorithm for estimating 3D finger pose including roll, pitch, and yaw from fingerprint images. 3D finger surface is first reconstructed based on sequential fingerprint images captured in enrollment, and given this 3D surface model, 3D finger pose of a test fingerprint is estimated by matching keypoints between the 2D image and 3D point cloud and minimizing the projection error. The proposed approach is a non-learning algorithm with good generalization ability and robustness in real applications. To evaluate the performance of our method, a dataset of fingerprint images with their corresponding ground truth 3D angles is collected. Experimental results on this dataset demonstrate the effectiveness of introducing reconstructed 3D finger surface shape in 3D finger pose estimation. The average absolute errors of three angles are 10.74 for roll, 8.25 for pitch, and 7.38 for yaw, respectively. Extensive experiments are also conducted to explore the impact of touching area size and gallery size on performance.

*Jianjiang Feng is the corresponding author

IUI '22, March 22–25, 2022, Helsinki, Finland

ACM ISBN 978-1-4503-9144-3/22/03…$15.00
https://doi.org/10.1145/3490099.3511123

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Computing methodologies** → *Artificial intelligence*.

KEYWORDS

3D finger pose, fingerprints, 3D surface reconstruction, keypoint matching

ACM Reference Format:
Yongjie Duan, Ke He, Jianjiang Feng, Jiwen Lu, and Jie Zhou. 2022. Estimating 3D Finger Pose via 2D-3D Fingerprint Matching. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3490099.3511123

1 INTRODUCTION

Over the past decades, touchscreens have become the widely used interaction techniques for smartphones, tablet computers, and other intelligent devices. When a finger touches the display, the capacitance is changed which can be measured by capacitive sensors, then touching points are calculated based on the measured capacitive changes [20]. Therefore, only 2D input is considered despite of the flexibility of human fingers movement. To extend the richness of user input vocabulary, commercial devices and recent researches have explored a wide range of novel interaction techniques. For example, tap-and-holding is distinguished with single click by leveraging time dimension [6]. Chording gestures are composed of sliding operation by single or multiple fingers [17]. *3D Touch* and hard press is provided by measuring the touch pressure [2, 28]. Besides, finger shape [24] and size [3], shear force [9], hand part [10], and finger angles [15, 20, 30, 35] are also applied to modify the touch input and alter the action.

Among these new interaction techniques, pressing orientation, i.e. 3D finger pose while touching on displays, has attracted numerous attention because of its great manipulate potential and intuition for human interaction. Similar with the definition on airplane, three angles: roll, pitch, and yaw are used to describe 3D finger pose. Roll refers the angle around the finger's longitudinal axis, pitch around the lateral axis, and yaw around the vertical axis. Figure 1 shows the definition of these three 3D angles. Previous works have suggested that pressing angle can be used to enrich touch input vocabulary and provide auxiliary information for complicated touch interaction [5, 39]. With the assistance of finger angles, the accuracy of

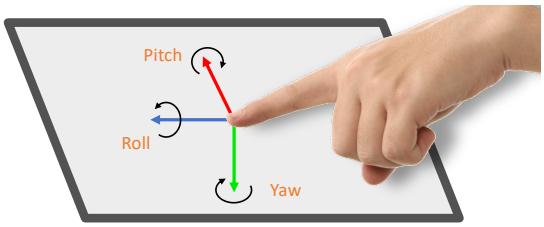

Figure 1: The definition of finger pose, which is described by three angles: roll, pitch, and yaw, while pressing on a screen.

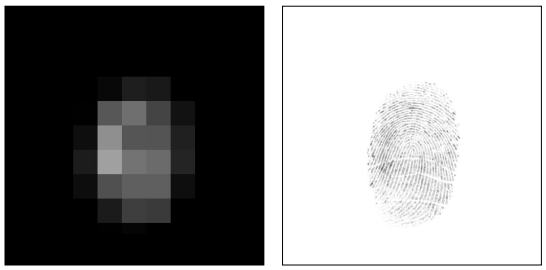

Figure 2: Examples of capacitive image (simulated) and fingerprint image pair.

touch inputs is improved [14], and the manipulation of 3D models or 2D images on smartphones or smartwatches which have limited screens becomes significantly convenient [35]. Some functions, such as adjusting font size, adjusting screen brightness, and adjusting brush width, can also be associated with touch angles [12]. Besides, the ergonomic constraints based on 3D finger pose are further explored as additional inputs while interacting with two hands [21]. Furthermore, the commonly used and unused range of finger orientation for different fingers while touching also provide insights for future interaction techniques development [7, 13].

Despite the many benefits of 3D finger angles as interaction input, estimating 3D angles based on 2D images is still a very challenging task. Up to date, capacitive images which are of quite low resolution and information poor are utilized and thereby making the estimation accuracy of finger angles not satisfactory. To alleviate the lacking information, some additional sensors, e.g. the photosensitive device attached to fingernail or additional depth cameras, were employed to provide auxiliary information [15, 34]. However, utilizing these complicated additional devices for finger pose estimation is inconvenient for human interaction in daily applications. In fact, with the development of under-screen fingerprint sensing techniques, a new input modality, fingerprint image, for 3D finger pose estimation becomes feasible [27, 37]. Figure 2 shows an example of comparison between capacitive image and fingerprint image. Different from capacitive images, relative positions of current touch area, which also indicates the 3D finger pose, are embedded in fingerprint images. In other words, with the finger pose changing, the pressed fingerprint image changes correspondingly. Furthermore, the deformation pattern of different fingers pressing on displays is intuitively different, making ridge patterns and minutiae contained in fingerprint images, which are fundamental features in fingerprint recognition, the very promising information for estimating all three finger angles. Previous researches have found that different fingers prefer different orientations and different ranges of angle while interacting [7], e.g. thumbs prefer to change pitch and roll angles while using smartphones with single hand, while pitch and yaw are convenient for the other fingers (such as index, middle, ring, and little). However, estimating roll angle in capacitive images, which are not sensitive to rolling, is rarely explored in previous studies.

In this paper, we proposed a non-learning framework to estimate 3D finger pose based on reconstructed 3D finger surface. Different from previous algorithms, we estimate all three finger angels, i.e. roll, pitch, and yaw, from fingerprint images at the same time. Given the accurately estimated finger angles, the poses of finger while touching displays can be fully described, which enriches the input vocabulary on touchscreens significantly and makes the interactions, e.g. manipulate 3D objects, more intuitive and convenient. To evaluate the proposed method, we collected a dataset consisting of 54 fingers from 9 subjects. For each finger, two fingerprint pressing sequences and the corresponding ground truth values of finger pose were collected with an interval over 6 months for enrollment and evaluation respectively. Based on the pre-enrolled fingerprint sequences and finger angles, we reconstruct the 3D surface shape for each finger and estimate three finger angles by matching 2D-3D keypoints and minimizing projection errors. Experimental results demonstrate the effectiveness of our method compared with previous capacitive images based algorithms. Since the proposed algorithm is developed without learning process, numerous training data with finger pose ground truth is not required and our method performs reliably and robustly, especially for limited user gallery size and sensors with small touching area.

2 RELATED WORKS

According to the sensing devices, conventional algorithms are classified into several categories: based on capacitive images, auxiliary devices, and fingerprint images. An overview of previous researches is shown in Table 1.

Touchscreen is widely used in various pervasive commercial devices, and capacitive images obtained from touchscreen, which describe the electrical capacitance changes when an object touching the display screen, have become the most common modality for finger pose estimation. Intuitively, with different finger poses, the silhouette and intensity distribution of touching area is different. This property is directly employed in finger pose estimation. For example, Wang et al. [32] estimated yaw angle using the shape of touching area in capacitive images. Apart from silhouette of touching area, Zaliva et al. [39] extracted several additional features, e.g. intensity, centroids, and asymmetry of shape, to compute 3D finger angles for gesture recognition. Similarly, in [35], 42 features including the ellipsoid's orientation and magnitude were extracted by assuming the shape of touching area in capacitive image as "comet", and then a Gaussian regression model was employed to predict pitch, yaw, and roll (while the estimation performance of roll angle is not reported). With the development of deep learning, Mayer et al. [20] adopted convolution neural networks (CNN) to predict finger pose based on capacitive images with 3D finger angels ground truth captured by an optical tracking system, and the estimation

Table 1: The overview of previous finger pose estimation algorithms. The results in "Errors" column are reported errors in their papers respectively. Mean absolute error (MAE) is the default metric.

Method	Year	Sensor	Ground Truth	Angles			Errors		
				Roll	Pitch	Yaw	Roll	Pitch	Yaw
Wang et al. [32]	2009	touchscreen	-			✓	-	-	-
Dang and André [5]	2011	touchscreen	-			✓	-	-	-
Zaliva [39]	2012	touchscreen	-	✓	✓	✓	-	-	-
Xiao et al. [35]	2015	touchscreen	plastic stands with specific angles		✓	✓	-	9.7	26.8
Mayer et al. [20]	2017	touchscreen	optical tracker		✓	✓	-	10.0	18.3
Rogers et al. [29]	2011	capacitive sensor array	-		✓	✓	-	-	-
Watanable et al. [34]	2012	photosensitive device	-	✓	✓	✓	-	-	-
Kratz et al. [15]	2013	depth camera	-		✓	✓	-	-	-
Mayer et al. [22]	2017	depth camera	multiple cameras		✓	✓	-	15.4*	12.2*
Holz and Baudisch [14]	2010	optical fingerprint scanner	optical tracker	✓	✓	✓	-	-	-
Su et al. [31]	2016	inking method	manual annotation			✓	-	-	6.65
Ouyang et al. [25]	2017	inking method	manual annotation			✓	-	-	2.65
Yin et al. [38]	2020	inking method	manual annotation			✓	-	-	1.45
Ours	-	optical fingerprint scanner	IMU, optical tracker	✓	✓	✓	10.74	8.25	7.38

* Root mean square error, RMSE

accuracy is further improved benefiting from the powerful feature extraction ability of deep neural networks. Due to the usage of capacitive images with low resolution and poor information, however, finger pose estimation performance is limited in these approaches. Furthermore, capacitive images are not sensitive to the roll angle while rolling is a convenient action for most fingers, therefore, roll angle is rarely estimated in these abovementioned methods.

To alleviate the lack of information, Rogers et al. [29] first attempted to use a set of capacitive sensors which were placed at lattices to track 2D touching position and 3D finger angels at the same time. A camera was attached to the fingernail in [34], and the luminance was measured to estimate current pitch and yaw angles, while the mounted camera constrains the nature range of finger orientation. Kratz et al. [15] and Mayer et al. [22] employed additional depth cameras to capture point clouds of fingers while touching on displays, then pitch and yaw angles were predicted based on the collected point clouds. These approaches show that 3D finger pose can be estimated by using sensors specially designed for this task. However, the application scenarios are constrained due to the difficulty and costs of adding additional sensors in smartphones or tablet computers for real applications.

More recently, fingerprint images have also been utilized in human interaction systems. Gust [8] calculated relative motion between two fingerprint images by analyzing optical flow. Holz and Baudisch [14] first proposed to use fingerprint images from a RidgePad prototype as an auxiliary information for rectifying touching points. In this method, a set of fingerprint images with corresponding finger angles were enrolled as finger specific gallery, and then by computing the similarity scores between input fingerprint and gallery fingerprint images, top-k matches were selected and merged to predicted finger angles of the input image. 3D finger pose is available though, the accuracy of pose estimation is not reported since the predicted finger angles are used for touching points rectifying eventually. Besides, estimation performance and time consuming are conflicting for this method: higher performance is achieved using pre-enrolled gallery with large size and wide range, while consuming more time in enrollment and testing.

In the field of fingerprint recognition, estimation of 2D fingerprint pose, the yaw angle, is also an important task. Various approaches have been proposed to predict fingerprint pose, e.g. Hough-voting [36], SVM classifier [31], and deep learning [25, 38]. Note that fingerprints used in these researches are rolled fingerprints typically captured in forensic applications that contain more information compared with plain fingerprints used in daily applications. And the range of yaw angle is limited in these fingerprint datasets collected in controlled environment, therefore, the estimation error is significantly small as shown in Table 1. Compared with 2D fingerprint images, 3D fingerprints have drawn increasing attention because of its superiority on the wider effective area and incorporation of 3D geometry information recently. However, up to date, reconstructing 3D fingerprints are always performed in contactless manners, e.g. reconstructing based on structured light illumination [33], shape-from-silhouette [26], and multi-view cameras [16, 18]. Obviously, reconstructing 3D fingerprint using contactless images or point clouds is not feasible for mobile devices where contact based fingerprint images are used.

3 DATA ACQUISITION

Since there is no public under-screen fingerprint dataset for 3D finger pose estimation, the first step in our study is to collect a new database for developing and evaluating our algorithm, which consists of fingerprint images with corresponding 3D finger angles ground truth. The overall acquisition system, shown in Figure 3, consists of an optical fingerprint scanner, an inertial measurement unit (IMU), an optical tracking camera, and some reflective markers for tracking. The optical fingerprint scanner[1] is utilized to capture fingerprint pressing sequences (video). The IMU[2] is attached to fingernail to record 3D finger angles while constructing finger specific gallery. And the optical tracking camera[3] is utilized to record 3D finger angles as ground truth for evaluation. The reason for using two different pose sensing devices is that utilizing optical tracking system in enrollment is not practical because of high cost and large size. Instead, using a small IMU is a more realistic and acceptable option.

[1] http://www.dotutech.com/en/pro_d.php?id=3
[2] https://www.wit-motion.com/9-axis/witmotion-jy901-ttl.html
[3] https://www.ps-tech.com/products-pst-iris

Figure 3: The whole data acquisition system. From left to right: optical fingerprint scanner with inertial measurement unit (IMU) and with optical tracking system. Reflective markers are used to track and determine finger pose.

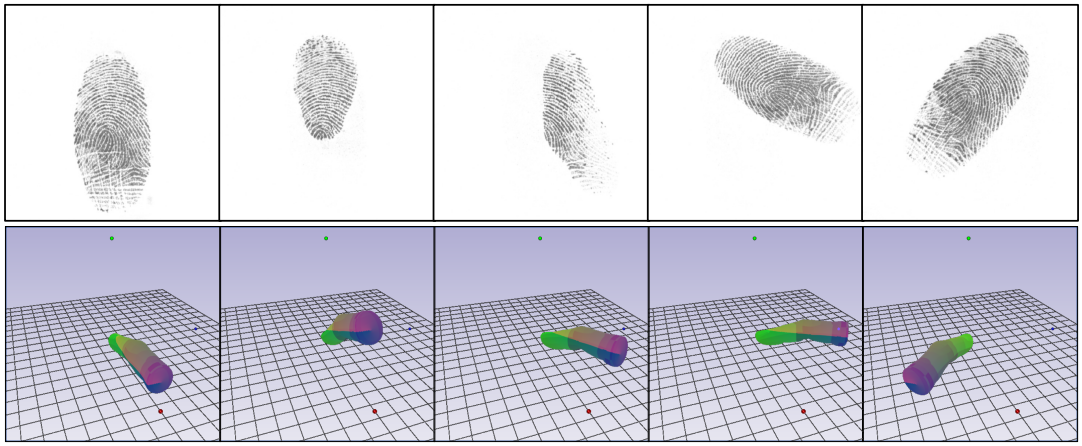

Figure 4: Some examples collected. The top row shows the fingerprint images and the bottom row displays the corresponding finger pose rendered in a 3D scene.

Totally 9 volunteers (9 men with average age of 25) were invited to participate in the data acquisition procedure. And 6 frequently used fingers including thumb, index, and middle from both left and right hands for each subject were captured. Fingerprint pressing sequences (video) as well as the corresponding 3D finger pose for each frame were collected by rolling fingers on the fingerprint scanner. Note that the acquisition process was applied twice with an interval over 6 months.

- Data collected in the first round was used as pre-enrolled gallery in our study. An IMU were attached to the finger to record 3D poses (see the left in Figure 3). Participants were advised to touch the fingerprint scanner by rolling their fingers from one side to the other side, i.e. mainly roll angle is changing. Eventually, a rolled-like fingerprint can be easily obtained by mosaicing these fingerprint images.
- Data collected in the second round was used for evaluation. A small stent with reflective markers were fixed on the finger, and finger pose was determined by calculating the relative transformation by tracking the reflective markers on the stent and fingerprint scanner respectively (see the right in Figure 3). In a comfortable way, participants were asked to change roll, pitch, and yaw angles of their fingers steadily to ensure a wide range for each angle, which is different from the first round.

In this way, image intensity and deformation pattern are different between two collections, therefore the generalization and robustness of finger pose estimation algorithms are able to be evaluated. The initial pose of each finger is aligned before capturing based on the data acquisition software developed by ourselves.

The size of captured fingerprint images is 800×750 pixels with the resolution of 500 ppi. The optical tracking system contains 4 reflective markers on the stent and 5 additional markers on the fingerprint scanner (see the right Figure 3). For IMU and optical tracking system, the sample rate of images and finger angles are set at the same frequency 50 Hz. Besides, the sequential images are not saved to disk until all frames have been collected.

Finally, a total of 23,099 fingerprint images and corresponding finger poses ground truth were collected. Some examples are shown in Figure 4.

4 METHODS

We propose a finger pose estimation algorithm based on fingerprint images using 3D surface reconstruction, keypoint matching, and projection reversing. To eliminate the redundant information contained in all fingerprint images during enrollment, 3D surface is reconstructed to represent the curvature of finger surface. Then a keypoint matching approach is applied to explore the dense registration between current frame and the reconstructed 3D model. Finally, finger pose is estimated by resolving the projection matrix based on the keypoints correspondence abovementioned. The overall flowchart of our method is illuminated in Figure 5.

4.1 Finger Surface Reconstruction

As demonstrated in [14], a finger specific gallery with many enrolled fingerprint images contains sufficient information for accurate pose estimation, while resulting redundant searching. However, a small size gallery might degrade estimation precision due to sparse distribution in 3D pose space. Therefore, to utilize information within fingerprint images more efficiently, we merge all fingerprint images and corresponding 3D finger poses within a finger specific gallery to reconstruct 3D finger surface, which can be used for pose estimation subsequently. For each frame, a coherence based fingerprint segmentation algorithm [1] is first applied to exclude background, and followed by a surface normals estimation to merge sequential information. Finally, given the surface normals, trapezoidal numerical integration is adopted to reconstruct finger surface.

Surface normal vectors estimation. Generally, fingerprint image plane is the tangent plane of finger surface in extreme circumstance that only one point within the touching area. Considering the skin deformation caused by pressing, therefore, a dense map of normal vectors, which plays a similar role as depth map, is required to describe the curvature of finger surface. Intuitively, image intensity is different when the touching pressure varies, thereby making it a rational assumption that curvature of finger skin $\boldsymbol{I}_{curv}$ is proportional to the touching pressure $\boldsymbol{I}_{press}$, i.e. intensity distribution $\boldsymbol{I}_{distrib}$

$$\boldsymbol{I}_{curv} \propto \boldsymbol{I}_{press} = \lambda \boldsymbol{I}_{distrib}.$$

λ is affected by various factors, e.g. finger shape and skin conditions, therefore, it should be different for different touching sequences. In depth-like map, according to the equation of curved surface, 3D surface normals are equivalent to 2D gradient vectors

$$\begin{cases} \boldsymbol{g}_x = (1, \nabla_x f)^{\mathrm{T}} \\ \boldsymbol{g}_y = (1, \nabla_y f)^{\mathrm{T}} \end{cases} \implies \boldsymbol{n} = (-\nabla_x f, -\nabla_y f, 1)^{\mathrm{T}}, \tag{1}$$

where f denotes the value in pressure map abovementioned, $\boldsymbol{n}$ denotes the surface normal, $\boldsymbol{g}_x$ and $\boldsymbol{g}_y$ denotes the gradient of X- and Y- direction respectively.

Considering the constraints of adjacent frames that the calculated surface normal at the same position are supposed to be paralleled, all frames are first transformed to a standard pose and then the surface normals estimation is optimized on the whole aligned sequence to calculate the optimal ratio $\{\lambda_k | k = 1, \ldots, K\}$ for each frame

$$\min_{\lambda_k > 0} \sum_{k=1}^{K} \left\| \lambda_k \boldsymbol{n}'_k - \overline{\boldsymbol{n}'} \right\|_2^2, \qquad \overline{\boldsymbol{n}'} = \frac{1}{K} \sum_{k=1}^{K} \lambda_k \boldsymbol{n}'_k \tag{2}$$

where K is the number of frames within a sequence, and $\boldsymbol{n}'$ is the aligned surface normal. Note that only points within the fingerprint segmentation mask are taken into account, and finally, all surface normals are calculated based on the optimized ratios.

Trapezoidal numerical integration. Given the estimated surface normals, we reconstruct finger surface by reversing the conversion in Equation 1

$$\boldsymbol{n} = (n_x, n_y, n_z)^{\mathrm{T}} \implies \begin{cases} \boldsymbol{g}_x = (n_z, -n_y)^{\mathrm{T}} \\ \boldsymbol{g}_y = (n_z, -n_x)^{\mathrm{T}} \end{cases}. \tag{3}$$

We select the deepest point as the origin where the surface normal is parallel to $\boldsymbol{z} = (0, 0, 1)^{\mathrm{T}}$. A trapezoidal numerical integration, i.e. integrating along X- direction first and then Y- or Y- direction first and then X-, is followed to reconstruct finger surface using the converted gradient vectors in Equation 3. Apart from the point cloud of finger surface, a point-to-point correspondence map to the mosaic fingerprint is obtained at the same time. Figure 6(a) presents several gallery images, and its reconstructed finger surface as well as the corresponding mosaic fingerprint is shown in 6(b).

4.2 Pose Estimation

Similar with the stereo vision that observed object features change obviously with different perspectives, the number and topology of keypoints also change when fingers touching with different pose. Based on this idea, we propose a 3D finger pose estimation algorithm by reversing the projection process, i.e. estimating the optimal projection parameters based on an orthogonal projection assumption in capturing fingerprint image.

Sparse keypoint matching. Firstly, keypoints correspondence between test fingerprint image and the mosaic fingerprint image is explored for a sparse initial registration. In [14], the widely used keypoint and descriptor extraction algorithm, speed up robust features (SURF), is utilized to calculate the similarity between test image and enrolled image. However, due to various skin conditions, matching performance between images collected with large interval is unsatisfactory. Consequently, minutiae, the most prominent ridge characteristics, are utilized in this study. The two main categories of minutiae, endpoints and bifurcations, are generally stable and robust to fingerprint impression conditions [19]. We use a commercial fingerprint preprocessing software VeriFinger to extract minutiae from fingerprints in this paper [23]. Figure 10 shows an example of fingerprint with two types of minutiae. A fixed-length descriptor, minutia cylinder code (MCC) [4], is utilized to represent the extracted minutiae. Then spectral clustering algorithm is employed to obtain the keypoint matching pairs and followed by a random sample consensus (RANSAC) algorithm to remove outliers. Figure 7(a) shows an example of keypoint matching after RANSAC correction.

Dense spreading. Normally, minutiae in fingerprint are not evenly distributed (see Figure 10), making the matching pairs obtained previously infeasible for pose estimation because of the confusion caused by sparse minutiae distribution, i.e. two test fingerprint images obtained under different finger poses might contain similar (even exactly the same) matching pairs, making it not trivial to estimate finger pose due to the dominance of similar matching pairs. Therefore, a dense spreading process is applied to expand the

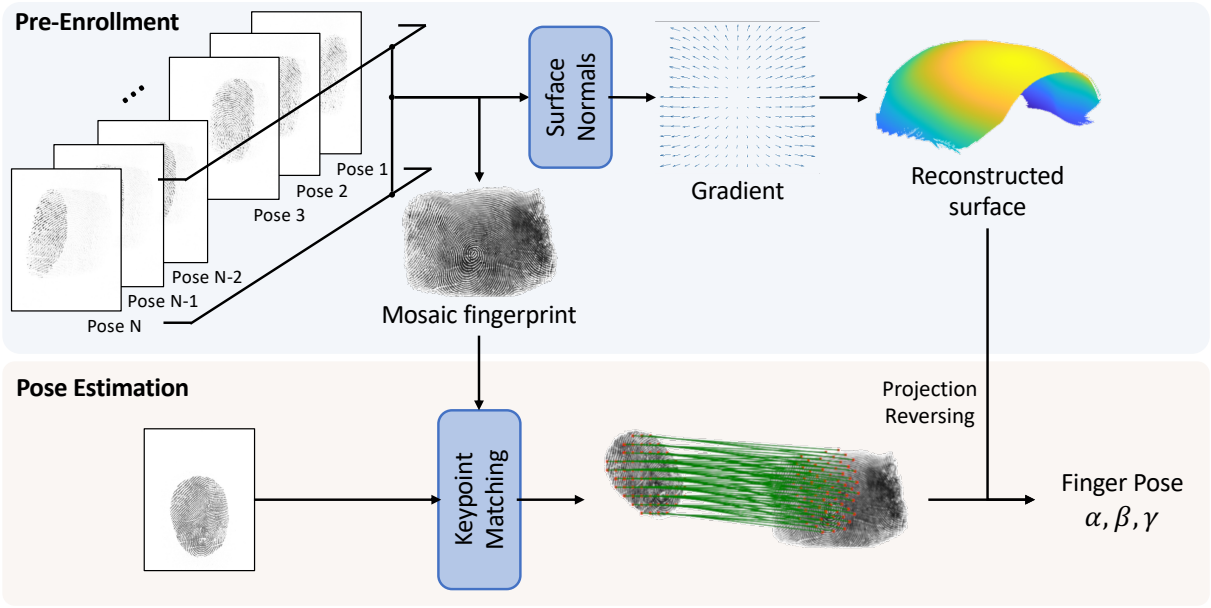

Figure 5: The schematic illustration of our algorithm. The finger pose is estimated by minimizing the 2D projection error of the corresponding 3D keypoints.

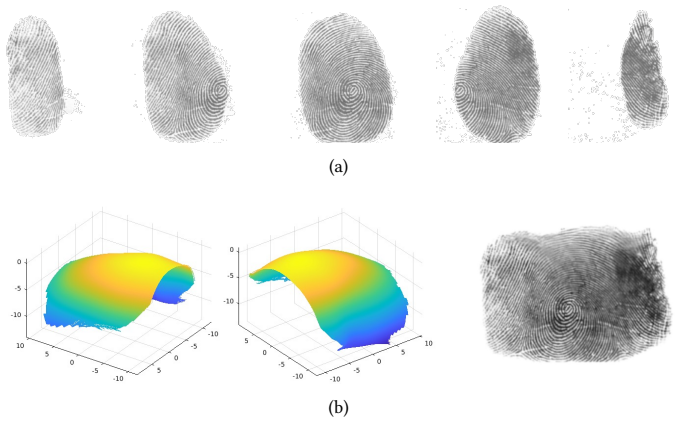

Figure 6: An example of finger surface reconstruction. (a) shows examples of gallery images, and (b) is the reconstructed 3D finger surface and corresponding mosaic fingerprint.

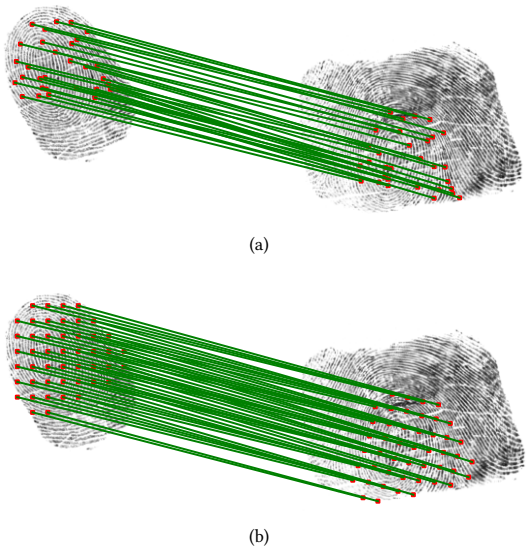

Figure 7: Keypoint matching in 3D pose estimation. (a) is an example of sparse keypoint matching, and (b) after the dense spreading.

matching pairs that lie on lattices while maintaining the correct matching transformation. Specifically, thin plate spline (TPS) is implemented to compute the dense correspondence between two fingerprint images, and points outside of fingerprint foreground mask are removed. An example of dense correspondence is illuminated in Figure 7(b).

Projection reversing. Given the point-to-point correspondence map abovementioned, dense matching pairs between 2D image and 3D model is further calculated and followed by a projection parameters prediction. Denote $\boldsymbol{p}_i = (u_i, v_i)$ and $\boldsymbol{P}_i = (x_i, y_i, z_i)$ as the i-th matching point in 2D image and 3D model respectively, and minimize the projection errors

$$\min_{s,\alpha,\beta,\gamma,t_x,t_y} \sum_{i=1}^{N} \left[(u_i - s\boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{P}_i - t_x)^2 + (v_i - s\boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{P}_i - t_y)^2 \right], \quad (4)$$

where $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ indicate the first and second row of rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{bmatrix} = \boldsymbol{R}(\gamma) \cdot \boldsymbol{R}(\beta) \cdot \boldsymbol{R}(\alpha).$$

s, α, β, γ, t_x, and t_y are projection parameters to be optimized which denote scale factor, roll, pitch, yaw angle, translation along X- direction, translation along Y- direction respectively.

4.3 Boosting

Considering the time constraints in human-computer interactions, it is not feasible to use a complicated minutiae extraction algorithm, i.e. VeriFinger SDK which is developed for robust and accurate fingerprint recognition under various complicated conditions. Therefore, a boosting version is proposed by combining the strengths of minutiae and SURF. Specifically, in test phase, minutiae based matching is performed between keyframe and mosaic fingerprint, while SURF based matching is performed between adjacent frames. In this way, robustness and performance of keypoint matching is ensured by minutiae based method. Then each frame is matched with its last adjacent frame based on SURF, which is more efficient than minutiae, and the matching correspondence with mosaic fingerprint is further obtained, which is used for projection error optimization. Besides, every frame is also matched with the last keyframe by SURF based matching algorithm and when the similarity score is lower than a specific threshold, a new keyframe is selected. Two separate threads are created for minutiae-based low-frequency matching and SURF-based high-frequency matching respectively to boost the estimation process.

5 EXPERIMENTS

To evaluate the performance of our proposed method, experiments on test fingerprint database are performed. As mentioned in Section 3, we collected a dataset with a total of 54 fingers. For each finger, two fingerprint image sequences with corresponding finger poses were collected twice with an interval over 6 months. The first collection, with IMU attaching to the fingernail to capture finger poses, is used as enrollment database for finger specific gallery conduction, i.e. reconstructing 3D finger surface in our method. While the second collection using optical tracking system is for evaluation. The detailed statistics of two dataset are summarized in Table 2. Furthermore, fingerprints are centered and cropped (the cropped image size is 480×480) using fingerprint segmentation mask to remove the position information, which may be correlated with pose, but should be avoided in pose estimation.

5.1 Baseline Methods

Since there are neither public implementations of finger pose estimation algorithms nor public fingerprint datasets for finger pose estimation, we reimplemented several algorithms and used our collected fingerprint images to perform evaluation. The same training (enrollment) and evaluation dataset are used for all baseline methods and the proposed approach. Note that for those non-enrollment approaches, images collected in the first round (enrollment dataset) are used for training, and images from the same finger collected in the second round (evaluation dataset) are used for evaluation, therefore the evaluation is relatively easier compared with enrollment-based methods.

- First, we compare with a multiple features based algorithm [35]. Since touching point is one of the outputs of capacitive sensors and not available in our fingerprint scanner, and it cannot be replaced by the moment of image intensity, which is defined as the centroid of ellipsoid in [35], we implemented this method without touching point related features, e.g. the distance and angle between touching point and the ellipsoid centroid. The rest features are extracted from the collected fingerprint images for training and evaluating the Gaussian process regressor algorithm.
- Deep learning has drawn lots of attention in object classification and regression due to its powerful feature extraction ability, therefore, we reimplement a CNN based model proposed by Mayer et al. [20]. Considering that the algorithm is developed for capacitive images originally, experiments taking two types of input images (fingerprint images and simulated capacitive images) are performed respectively to make a comprehensive comparison. The capacitive images are simulated by down-sampling the original fingerprint images to match the general resolution of capacitive images (~4 mm pitch). An example of simulated capacitive image is shown in Figure 2. Considering it is unfair to feed high resolution fingerprint images to the shallow network in [20], which is designed for extremely low resolution capacitive images. we extended the network backbone to ResNet-50 [11], which consists of more parameters and deeper layers than the CNN model in [20]. Similarly, for this deeper network model, both two types of input images abovementioned are also employed for training and evaluation respectively. Images from enrollment dataset are split for training and validation with a ratio of 9 to 1. Data augmentation is performed on training images, including randomly cropping, rotation, and translation. We trained the shallow model in [20] from scratch and parameters pre-trained on ImageNet are used to initialize ResNet-50. The same loss functions and hyperparameter settings in [20] are utilized. We implement the deep network model in [20] and its extended model (ResNet-50) on PyTorch with NVIDIA GeForce 2080Ti.
- Holz and Baudisch's approach [14] is an enrollment-based algorithm where the final finger pose is predicted by exhaustive searching in a finger specific gallery, and SURF based feature extraction and matching algorithm is utilized to determine image similarities. Originally, this method was designed for touching point rectifying, although, we found that it can be used for finger pose estimation intuitively. The finger gallery dataset used for searching is the same as the proposed method. Instead of SURF, we employ minutiae based algorithm to extract image descriptors and calculate image similarities, since SURF features are not robust to the obvious difference between images from two collections with large interval. Besides, based on keypoint matching pairs extracted by minutiae based matching algorithm, an

additional yaw angle rotation offset is calculated to improve the yaw estimation accuracy, which is not introduced in [14].

5.2 Performance Analysis

In Table 3, we compare the proposed method with baseline approaches on the collected evaluation dataset. As shown in the table, our non-learning method performs better than the other algorithms with or without pre-enrollment. Note that the estimation error of pitch angle is small for all fingerprint based algorithms, which is reasonable since the distribution variance of pitch angle is limited (see Table 2) in evaluation dataset. Besides, we found that the two deep learning methods, Mayer et al. [20] and its deeper extension ResNet-50, have a higher estimation accuracy using fingerprint images compared with simulated capacitive images, which demonstrates the importance of introducing input images with higher image resolution, e.g. fingerprint images. Figure 9 shows the error distributions of our method given its ground truth value of three angles. Compared with the intrinsic distribution of the evaluation dataset, the estimation error is reasonable since keypoint matching might perform unsatisfactorily due to very small number of minutiae for large pitch or roll angles. Several additional explorations on fingerprint size and gallery size are shown as follows.

Errors for different visible sizes. Employing large size under-screen fingerprint sensors on commercial devices is expensive. Therefore, the pose estimation performance on the limited area matters in real applications. To examine the performance of our method on those sensors with limited size, we conducted three experiments where fingerprint images are randomly cropped into 80%, 60%, and 40% visible size respectively. The resulting fingerprint images are shown in Figure 10. The evaluation results and comparison with ResNet-50, the best-performing deep learning method without enrollment, are listed in Table 4. We find the estimation accuracy of deep model degrades significantly with the size of fingerprint area decreasing. It is reasonable because limited ridge pattern information is available and silhouette information is lost when the size of visible fingerprint area decreases. Benefiting from the non-learning scheme, the proposed method has a slight performance declining compared with ResNet-50, which demonstrates the robustness of our method on visible fingerprint area size.

Errors for different gallery sizes. For enrollment-based algorithms, the representativeness of gallery has a significant impact on the performance, while time-consuming and complicated enrollment should be avoided in real applications. To examine the performance of our method with different gallery sizes, we decreased the finger gallery size to 10 and 3 respectively. The gallery images were selected evenly making the range of angles invariable compared with the whole finger gallery. Meanwhile, additional two experiments were further conducted where the gallery images were selected excluding the images with large angles, i.e. the absolute value of roll angle greater than 45 and 30 degrees respectively, therefore, the range of angles in finger gallery is limited, which is common in real applications. Note that 10 images were evenly selected from the finger gallery with limited range of angles. The estimation errors and comparison with Holz and Baudisch method [14] are shown in Table 5 and 6. Apparently, in our approach, the quality of 3D reconstruction model and mosaic fingerprint image get worse with fewer gallery images, therefore, the performance degrades slightly with the gallery size decreasing. Compared with Holz and Baudisch method [14], only one mosaic fingerprint and a 3D reconstructed finger surface are utilized in our approach, and the information within enrolled fingerprint images is fully utilized by 3D surface reconstruction. Consequently, less time is required for enrollment with a smaller gallery size while not declining the estimation performance significantly in our algorithm, and image matching is applied once between test fingerprint and mosaic fingerprint, thus reducing computation and memory consumption in test phase. Besides, the weighted averaging in [14] might not be reasonable when the 3D pose of fingerprint is out of the range of angles in gallery, which is alleviated in our approach based on the 3D information of reconstructed finger surface.

The potential applications of pitch and yaw angles have been discussed in previous studies, while as demonstrated in [7], different orientation range is preferred for different fingers. Only two angles, pitch and yaw, might not be sufficient in several constrained scenarios, e.g. rolling and pitching a thumb is relatively convenient compared to rotating the finger while using smartphones with single hand, therefore, an accurately estimated roll angle of thumb finger might provide more possible interactions for users in daily applications.

6 LIMITATIONS

Although a good performance is achieved in this preliminary study of estimating finger pose using fingerprint images, the current work has following limitations.

As mentioned in Section 3 and 4.1, a simple rolling is adopted to construct the finger specific gallery, where the fingertip area is ignored. Therefore, the performance is unsatisfactory when the pitch angle is large because of fewer matching points detected. While this problem can be alleviated by an additional rolling where fingers are tilted to capture fingertip area, then both two 3D finger surfaces reconstructed with normal and tilted finger, are utilized to determine the finger pose of a test fingerprint.

Different from the methods estimating finger pose directly, a pre-enrollment is required in the proposed approach. And only fingerprint images pressed by the enrolled fingers can be utilized to estimate finger pose, which means enrolling with index finger while estimating the finger pose of thumb finger accurately is infeasible. However, considering that there are already fingerprint enrollments in many smartphones, tablet computers, and other commercial devices, users are familiar with the process of fingerprint enrollment. Besides, a finger specific enrollment might help to improve the estimation accuracy because of the customized model. Although IMU is required to fixed a finger during enrollment which might not be convenient in practical, it can be alleviated by applying automatic relative pose estimation between adjacent pre-enrolled frames, and even simply, users could be trusted to approximate the desired finger pose as best as they can by following a guidance shown on the screen, since relative pose is sufficient for reconstructing the 3D finger surface.

Besides, fingerprint images in our dataset were collected using a traditional optical fingerprint scanner instead of the under-screen fingerprint sensors in mobile phones, whose image quality might

Table 2: Distributions of three angles in our dataset. Left is for enrollment and right for evaluation dataset. '#SN' indicates the number of frames within a sequence.

	Roll	Pitch	Yaw	#SN
Min	-74.60	-45.79	-31.23	20
Max	73.70	16.41	55.94	74
Mean	-3.02	-17.57	5.88	37
Std	41.43	11.46	11.20	13

	Roll	Pitch	Yaw	#SN
Min	-59.92	-45.02	-74.97	95
Max	60.00	-0.08	75.00	543
Mean	-4.34	-25.77	-0.38	316
Std	28.14	9.25	44.01	114

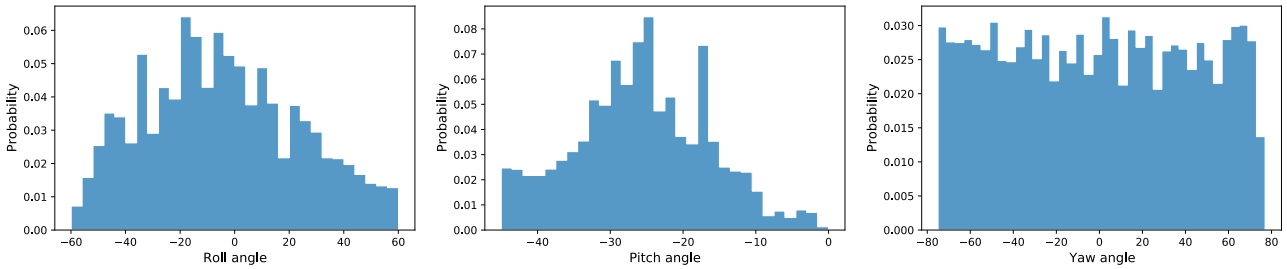

Figure 8: Distributions of three angles in evaluation dataset.

Table 3: Quantitative results for pose estimation algorithms. Errors are reported in degrees.

Algorithm	Enroll	Roll		Pitch		Yaw		Overall	
		MAE	SD	MAE	SD	MAE	SD	MAE	SD
Xiao et al. [35]	No	24.91	19.64	11.71	9.82	34.84	15.22	24.11	17.92
Mayer et al. [20][1]	No	28.01	19.15	11.82	8.07	34.19	20.58	24.67	19.34
Mayer et al. [20]	No	23.58	17.14	9.02	6.59	28.44	21.78	20.34	18.40
ResNet-50[1]	No	26.58	20.15	10.87	7.51	29.29	34.39	22.25	24.78
ResNet-50	No	14.93	12.01	6.98	5.81	12.84	13.00	11.58	11.28
Holz and Baudisch [14]	Yes	11.22	10.57	9.04	7.76	7.99	10.49	9.42	9.79
Ours (boosting)	Yes	11.24	12.76	8.75	8.75	7.88	12.76	9.29	11.66
Ours	Yes	10.74	12.74	8.25	8.71	7.38	12.73	8.79	11.64

[1] Simulated capacitive images are used for training and testing

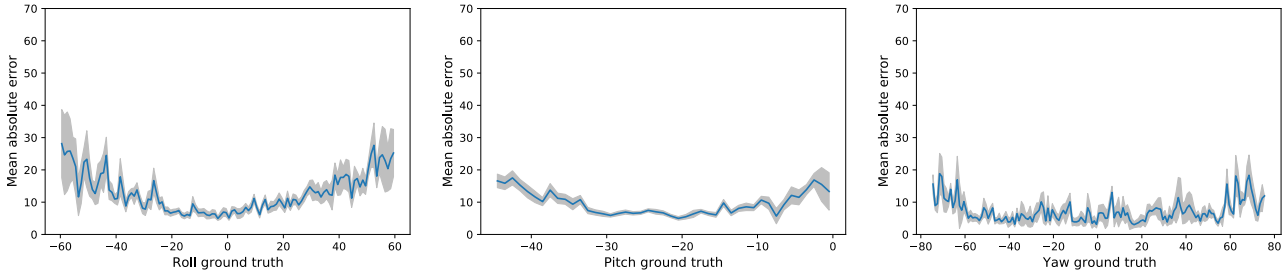

Figure 9: Error distributions of our method for three angles collected in evaluation dataset. The gray area indicates 95% confidence interval (CI).

be lower. Fingerprints of various factors are also not explored, e.g. wetness, dryness, gender, and ages, which means the quality of the collected fingerprint images is relatively high in this study. And we implement the proposed algorithm on a PC with i5-6500 CPU and 24GB RAM rather than mobile devices which have limited computation resources, therefore, more specific algorithm optimization and development are required for real applications.

Furthermore, fetching fingerprint images or extracted identity information, e.g. minutiae, from the Trusted Execution Environment (TEE) of smartphones is still a controversial issue and in conflict with current privacy policy for various third-party applications, not to mention the constrained computation and memory resources on mobile devices.

The potential applications of 3D finger pose has been extensively discussed in previous studies [5, 14, 21, 35, 39], and deploying the

Table 4: Performance of different fingerprint visible area sizes. MAE is utilized for evaluation. The average number of matching pairs (ANMP) is reported to show the sparse keypoint matching performance.

		Roll	Pitch	Yaw	Overall	ANMP
ResNet-50	full size	14.93	6.98	12.84	11.58	-
	80%	22.21	8.17	17.90	16.09	-
	60%	27.79	9.80	25.77	21.12	-
	40%	30.58	11.26	35.50	25.78	-
Ours	full size	10.74	8.25	7.38	8.79	35.5
	80%	10.99	8.72	8.00	9.24	31.3
	60%	11.26	9.33	8.67	9.75	26.6
	40%	12.60	10.88	10.40	11.29	19.8

Table 5: Performance of different gallery sizes. MAE is utilized for evaluation. The average number of matching pairs (ANMP) is reported to show the sparse keypoint matching performance.

		Roll	Pitch	Yaw	Overall	ANMP
Holz and Baudisch [14]	All	11.22	9.04	7.99	9.42	37.2
	10	11.24	9.01	8.13	9.46	36.5
	3	12.54	8.99	9.35	10.29	32.8
Ours	All	10.74	8.25	7.38	8.79	35.5
	10	11.07	9.32	7.97	9.45	35.5
	3	11.67	9.19	8.27	9.71	35.4

Table 6: Performance of different ranges of roll angle in gallery. MAE is utilized for evaluation. The average number of matching pairs (ANMP) is reported to show the sparse keypoint matching performance.

		Roll	Pitch	Yaw	Overall	ANMP
Holz and Baudisch [14]	complete range	11.24	9.01	8.13	9.46	36.5
	$\leq 45^\circ$	15.78	8.91	9.91	11.53	36.2
	$\leq 30^\circ$	19.39	8.91	11.47	13.26	35.1
Ours	complete range	11.07	9.32	7.97	9.45	35.5
	$\leq 45^\circ$	14.99	10.13	9.44	11.52	34.5
	$\leq 30^\circ$	15.07	10.26	9.49	11.61	34.5

Figure 10: Fingerprint of various visible sizes. From left to right, full-sized, 80%**,** 60%**, and** 40% **images with visible minutiae.**

proposed algorithm to mobile phones is still infeasible due to above-mentioned reasons, therefore, user studies with possible applications based on the estimated 3D finger pose is not conducted in this paper.

7 CONCLUSION AND FUTURE WORK

In this paper, an enrollment based framework is proposed to estimate 3D finger pose using fingerprint images via 3D finger surface reconstruction, 2D-3D keypoint matching, and projection reversing. To explore the feasibility and evaluate the performance of our method, a dataset consisting of fingerprint images and the corresponding ground truth finger poses is collected. To utilize the information contained in pre-enrolled images, we reconstruct the 3D finger surface to merge all gallery images, then a 2D-3D keypoint matching followed by a projection error minimization is applied to obtain the finger pose prediction. The experimental results demonstrate the effectiveness of our approach, especially when the visible touching area is relatively small that contour information is lost and the representativeness of finger specific gallery is limited, which are important factors in real applications. The proposed work is a preliminary study for finger pose estimation using fingerprint images. There are still a couple of problems to be solved for real applications, which will be further explored in future work.

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] Asker M Bazen and Sabih H Gerez. 2001. Segmentation of fingerprint images. In *ProRISC 2001 Workshop on Circuits, Systems and Signal Processing*. 276–280.

[2] 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*. 1–6. https://doi.org/10.1145/3338286.3344389

[3] 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

[4] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. 2010. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (2010), 2128–2141. https://doi.org/10.1109/TPAMI.2010.52

[5] Chi Tai Dang and Elisabeth André. 2011. Usage and recognition of finger orientation for multi-touch tabletop interaction. In *IFIP Conference on Human-Computer Interaction*. Springer, Berlin, Heidelberg, 409–426.

[6] John Greer Elias, Wayne Carl Westerman, and Myra Mary Haggerty. 2010. Multi-touch gesture dictionary. U.S. Patent 7 840 912.

[7] Alix Goguey, Géry Casiez, Daniel Vogel, and Carl Gutwin. 2018. Characterizing finger pitch and roll orientation during atomic touch actions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3173574.3174163

[8] Lawrence Alan Gust. 2006. Compact optical pointing apparatus and method. U.S. Patent No. 7 102 617.

[9] Chris Harrison and Scott Hudson. 2012. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3149–3152. https://doi.org/10.1145/2207676.2208730

[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*. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[12] Kourtny M Hicks. 2016. Stylus sensitive device with stylus angle detection functionality. U.S. Patent No. 9 448 643.

[13] Eve Hoggan, John Williamson, Antti Oulasvirta, Miguel Nacenta, Per Ola Kristensson, and Anu Lehtiö. 2013. Multi-touch rotation gestures: Performance and ergonomics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3047–3050. https://doi.org/10.1145/2470654.2481423

[14] 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. https://doi.org/10.1145/1753326.1753413

[15] 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*. 223–230. https://doi.org/10.1145/2512349.2512824

[16] Ruggero Donida Labati, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti. 2015. Toward unconstrained fingerprint recognition: A fully touchless 3-D system based on two views on the move. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46, 2 (2015), 202–219. https://doi.org/10.1109/TSMC.2015.2423252

[17] G Julian Lepinski, Tovi Grossman, and George Fitzmaurice. 2010. The design and evaluation of multitouch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2233–2242. https://doi.org/10.1145/1753326.1753663

[18] Feng Liu and David Zhang. 2014. 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model. *Pattern Recognition* 47, 1 (2014), 178–193. https://doi.org/10.1016/j.patcog.2013.06.009

[19] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. 2009. *Handbook of fingerprint recognition*. Springer Science & Business Media. https://doi.org/10.1007/978-1-84882-254-2

[20] 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. https://doi.org/10.1145/3132272.3134130

[21] 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*. 1–9. https://doi.org/10.1145/3229434.3229444

[22] 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*. 1–8. https://doi.org/10.1145/3098279.3122125

[23] Neurotechnology. 2020. Verifinger SDK. https://www.neurotechnology.com/verifinger.html.

[24] Ian Oakley, Carina Lindahl, Khanh Le, DoYoung Lee, and MD Rasel Islam. 2016. The flat finger: Exploring area touches on smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 4238–4249. https://doi.org/10.1145/2858036.2858179

[25] Jiahong Ouyang, Jianjiang Feng, Jiwen Lu, Zhenhua Guo, and Jie Zhou. 2017. Fingerprint pose estimation based on faster R-CNN. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 268–276. https://doi.org/10.1109/BTAS.2017.8272707

[26] Geppy Parziale, Eva Diaz-Santana, and Rudolf Hauke. 2006. The surround imager tm: A multi-camera touchless device to acquire 3D rolled-equivalent fingerprints. In *International Conference on Biometrics*. Springer, 244–250. https://doi.org/10.1007/11608288_33

[27] Chang Peng, Mengyue Chen, and Xiaoning Jiang. 2021. Under-display ultrasonic fingerprint recognition with finger vessel imaging. *IEEE Sensors Journal* 21, 6 (2021), 7412–7419. https://doi.org/10.1109/JSEN.2021.3051975

[28] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure widgets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 487–494. https://doi.org/10.1145/985692.985754

[29] 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. https://doi.org/10.1145/1978942.1979318

[30] Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 927–936. https://doi.org/10.1145/1518701.1518843

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

[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*. 23–32. https://doi.org/10.1145/1622176.1622182

[33] Yongchang Wang, Laurence G Hassebrook, and Daniel L Lau. 2010. Data acquisition and processing of 3-D fingerprints. *IEEE Transactions on Information Forensics and Security* 5, 4 (2010), 750–760. https://doi.org/10.1109/TIFS.2010.2062177

[34] 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 *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 601–612. https://doi.org/10.1007/978-3-642-31401-8_53

[35] 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. https://doi.org/10.1145/2817721.2817737

[36] Xiao Yang, Jianjiang Feng, and Jie Zhou. 2014. Localized dictionaries based orientation field estimation for latent fingerprints. *IEEE transactions on Pattern Analysis and Machine Intelligence* 36, 5 (2014), 955–969. https://doi.org/10.1109/TPAMI.2013.184

[37] 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 (2021), 763–777. https://doi.org/10.1109/JSSC.2020.3042894

[38] Qihao Yin, Jianjiang Feng, Jiwen Lu, and Jie Zhou. 2020. Joint Estimation of Pose and Singular Points of Fingerprints. *IEEE Transactions on Information Forensics and Security* 16 (2020), 1467–1479. https://doi.org/10.1109/TIFS.2020.3036803

[39] 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, 359–364. https://doi.org/10.1109/ICARCV.2012.6485185