PrintShear: Shear Input Based on Fingerprint Deformation

JINYANG YU, Department of Automation, Tsinghua University, China
JIANJIANG FENG*, Department of Automation, Tsinghua University, China
JIE ZHOU, Department of Automation, Tsinghua University, China

Most touch-based input devices, such as touchscreens and touchpads, capture low-resolution capacitive images when a finger touches the device's surface. These devices only output the two-dimensional (2D) positions of contacting points, which are insufficient for complex control tasks, such as the manipulation of 3D objects. To expand the modalities of touch inputs, researchers have proposed a variety of techniques, including finger poses, chording gestures, touch pressure, etc. With the rapid development of fingerprint sensing technology, especially under-screen fingerprint sensors, it has become possible to generate input commands to control multiple degrees of freedom (DOF) at a time using fingerprint images. In this paper, we propose PrintShear, a shear input technique based on fingerprint deformation. Lateral, longitudinal and rotational deformations are extracted from fingerprint images and mapped to 3DOF control commands. Further DOF expansion can be achieved through recognition of the contact region of the touching finger. We conducted a 12-person user study to evaluate the performance of PrintShear on 3D docking tasks. Comparisons with other input methods demonstrated the superiority of our approach. Specifically, a 19.79% reduction in completion time was achieved compared with conventional touch input in a full 6DOF 3D object manipulation task.

CCS Concepts: • **Human-centered computing → Interaction techniques**.

Additional Key Words and Phrases: touch input, shear input, 3d object manipulation

ACM Reference Format:
Jinyang Yu, Jianjiang Feng, and Jie Zhou. 2023. PrintShear: Shear Input Based on Fingerprint Deformation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2, Article 81 (June 2023), 22 pages. https://doi.org/10.1145/3596257

1 INTRODUCTION

Capacitive sensing-based touch input has been widely used and researched for mobile devices for more than two decades [19]. Most applications only utilize the 2D positions of touch points converted from low- resolution images captured by the sensors. With the increasing need for accurate and efficient interaction techniques for complex tasks such as 3D object manipulation in computer-aided design (CAD), electronic games, virtual reality (VR), and augmented reality (AR), researchers have proposed various approaches to extend touch input. For example, finger pose has been studied to improve touch accuracy and provide additional input modalities [39, 52, 54]. Touch pressure [6], finger shape [42] and touch area [7] can be estimated from capacitive images which provide possibilities for richer interactions.

Shear input is one of the new modalities which has been explored by researchers [22, 30, 32, 34]. It refers to the use of a force vector parallel to the device surface as input [22]. Similar to touch pressure, the shear force can

*Corresponding author

Authors' addresses: Jinyang Yu, jy-yu20@mails.tsinghua.edu.cn, Department of Automation, Tsinghua University, Beijing, China; Jianjiang Feng, jfeng@tsinghua.edu.cn, Department of Automation, Tsinghua University, Beijing, China; Jie Zhou, jzhou@tsinghua.edu.cn, Department of Automation, Tsinghua University, Beijing, China.

2474-9567/2023/6-ART81
https://doi.org/10.1145/3596257

be distinguished from normal touch and used in concert with touch input. However, most shear input techniques use additional sensors or attachments to detect shear force which is inconvenient or expensive for mobile devices.

Thanks to the rapid development of under-screen fingerprint sensing technology, fingerprint images can be captured when a finger is pressed on the screen [44, 53]. Therefore, it becomes possible to use fingerprint images as a new input modality beyond user authentication. Different from low-resolution capacitive images, fingerprint images contain rich information about a finger, including clear ridges, minutiae, touch area, etc. It can be observed in Figure 1 that, when a shear force is created by a finger, friction will cause the contacting area on the fingertip to deform. Deformation in the center region (stick region) is small while that in the boundary region (slip region) is significant [32]. In this paper, we propose PrintShear, a shear input technique based on fingerprint deformation.

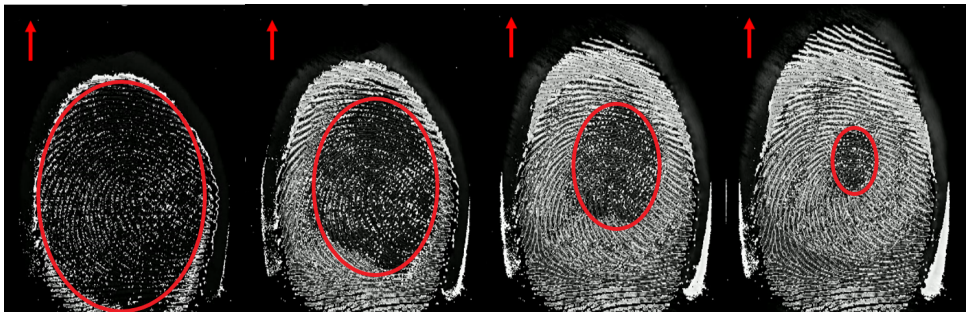


Fig. 1. Deformation of the fingertip under shear force. Images are created by taking differentials between the first frame and the following four frames in a fingerprint sequence. Red arrows indicate the direction of shear force and the regions inside the red circles represent the stick regions.

Unlike most shear input methods, which rely on additional sensors to measure shear force directly, we calculate skin deformations caused by shear force from fingerprint images. Feature points from adjacent frames in the captured fingerprint sequence are extracted and matched. Lateral, longitudinal and rotational deformations are then determined based on distance and the relative position between matched points. Deformations are finally mapped to 3DOF control commands and further DOF expansion can be achieved through recognition of the contact region of the touching finger. Two common multiple DOFs control includes 3DOF control, where a user controls the rotation (or translation) of an object or scene in the x-, y- and z-axis, and 6DOF control, where a user controls both rotation and translation in three axes. Since deformations are used as input, PrintShear only requires minimal movements of fingers and a small fingerprint sensing area, which is cost-effective for mobile devices. The proposed technique is a simple non-learning method, which requires only tens of milliseconds of processing time. We evaluated the performance of PrintShear on multiple 3D docking tasks in this study including a 2DOF translation task, a 3DOF rotation task and a full 6 DOF manipulation task. PrintShear outperformed conventional touch input in the 3DOF and 6DOF manipulations tasks, with a 14.65% and 19.79% reduction in completion time, respectively. To further explore the potential of PrintShear in desktop and laptop computers, comparisons were made with TrackPoint [46] and SpaceMouse[1], which are normally used for computers.

The main contributions of this work are:

- We propose PrintShear, a shear input technique based on fingerprint deformation. Lateral, longitudinal and rotational deformations are extracted from fingerprint images and mapped to 3DOF control commands. The modalities can be further extended when using different finger regions and more than 6 DOFs control can be achieved using a single finger.

[1]https://3dconnexion.com/us/spacemouse/

- We evaluate the deformation ability of different regions in different fingers and demonstrate the feasibility of PrintShear for each region.
- We conduct empirical experiments to evaluate the performance of PrintShear and compare it with other popular 3D object manipulation techniques and devices in both mobile and desktop computing devices.

2 RELATED WORK

Fingerprint deformation-based input shares properties of touch input and shear input. Fingerprint deformation has been studied from the perspective of fingerprint recognition.

2.1 Touch Input

Touch input is primarily based on 2D positions of touch points converted from low-resolution images captured by capacitive sensors. Tremendous work has been done to expand the modalities of touch input. Hancock et al. [20] proposed a Sticky Tools technique to use three-finger gestures for 6DOF object manipulation. It assigns single-finger dragging to control xy-translation, two-finger pinching for z-translation, and two-finger swiveling for z-rotation. Xy-rotation is performed using three fingers, where two fingers remain at fixed positions and the movement of the third finger determines the direction of rotation. Martinet et al. [38] also introduced a three-finger gesture, named DS3, which separates translation and rotation by examining the number of fingers that are in direct contact with the object. Liu et al. [35] further improved the Sticky Tools technique that uses two-finger gestures for 6DOF manipulation of 3D objects. Xy-rotation is performed using one finger to pin down at a fixed position and the other finger to rotate the object. A common disadvantage of gesture-based input is the occupation of screen space and occlusion of view, especially for screens of small size. Performance of gesture input also suffers from gesture recognition failures. For example, it is difficult to maintain a constant distance between two fingers when performing a panning gesture, which will cause a false pinching gesture to be recognized [21].

Finger pose has been widely studied to provide additional input modalities in recent years [25, 31, 39, 49, 52, 54]. Xiao et al. [31] proposed to use the shape of capacitive image to estimate yaw and pitch angles. Mayer et al. [39] used a convolutional neural network (CNN) to further reduce the errors of yaw and pitch angles. Zaliva [54] used multiple shape features, including area, average intensity, centroids, etc. to calculate three finger angles. Limited by the image quality, the accuracy of angle estimation from capacitive image is relatively low.

More recently, Duan et al. [15] proposed a 2D-3D fingerprint matching algorithm to estimate three finger angles. Fingerprint images and finger angles are recorded to construct a 3D surface model in an enrollment phase. Finger angles of a test fingerprint can be estimated by point matching between the 2D image and the 3D model. He et al. [25] proposed a deep neural network with multi-task learning to estimate three finger angles directly from fingerprints and achieved state-of-the-art (SOTA) accuracy. However, neither of the two methods conducts an empirical experiment on 3D object manipulation tasks. A major issue with angle estimation using fingerprints is that the accuracy relies on fingerprint quality. Fingerprint with low quality may lead to poor performance [25]. Using finger angle as input will also suffer from the jitter effect, which refers to unintentional fluctuations in movement [3]. Since the minimum angle error in the SOTA algorithm for finger angle estimation is 6.6 degrees, it will be difficult to complete a docking task with a tolerance of 3 degrees.

Other modalities such as touch pressure [6, 12, 13, 27], finger shape [42], touch area size [7] and different parts of hand [11, 23, 29, 33] have also been researched. These modalities are effective extensions for gesture input, but unable to provide enough DOFs for object manipulation tasks.

2.2 Shear Input

Shear input refers to the use of a force vector parallel to the device surface as input. Harrison and Hudson [22] created a prototype device by placing two analog joysticks underneath a touchscreen. A 2D shear force applied to the touchscreen will cause the joysticks to displace in response. Three movement modes can be realized using the device: a traditional drag mode where only position changes of touching points are detected; a push mode where only shear force is detected due to static friction; and a hard drag mode where both shear force and position changes are detected since shear force is created due to sliding friction. Five advanced interaction classes are also introduced by the authors. Xiao et al. [51] used a similar structure on a smartwatch and further enriched the available actions, including twist, tilt and click. Analog twist can be calculated by measurement of opposite movements of the two joysticks. Tilt is detected when a single joystick is pressed while click is detected when both joysticks are pressed. Heo and Lee [26] designed a sensing frame which is a mobile phone case-shaped frame with force sensing resistors attached to the bottom and side walls. The normal and shear force applied to the mobile device surface can be measured by the sensors. Force gestures are introduced using combinations of normal force, shear force and position movement. Lee et al. [34] used a similar device to investigate the user controllability of shear force considering hand pose and force direction. Huang [30] proposed ShearSheet, a rubber-mounted transparent sheet to enable shear input on a touchscreen. Relative movement between the sheet and touchscreen caused by shear force can be measured by the tiny conductive tape attached under the sheet and used as input. Nakai [41] et al. placed a transparent gel layer on top of a touchscreen and used the deformation of the gel layer to measure shear force. Shear force can be calculated as the product of finger displacement and spring ratio of the gel. A common disadvantage of shear input is that additional sensors or attachments are required to measure shear force which is inconvenient for mobile devices.

Work has been done to measure shear force indirectly without using additional sensors. Heo and Lee [28] proposed an indirect method to estimate shear force using the movement of contact areas. To create shear force, normal force must be applied to the surface. The size of the contact area reflects the normal force applied and is therefore used to detect a shear force event. The center of the contact area slightly changes when a shear force is applied and the displacement is roughly proportional to the magnitude of the shear force. To distinguish between shear (shear without sliding) and drag (shear with sliding), a speed threshold is used. Fast movement is classified as drag and slow movement is classified as shear. An experiment was carried out to test the feasibility of the proposed method and results showed that most errors were caused by shear force event detection failures. The authors made a good attempt to create shear input using capacitive images without help from additional sensors. However, using the size of the contact area to detect shear force events is relatively unreliable since the contact area is also dependent on finger pose and the location of the finger. In addition, using a speed threshold to distinguish shear from drag is not effective since fast shear and slow drag are both possible actions that cannot be separated using a speed threshold.

Kurita et al. [32] proposed a technique to detect the deformation of a fingertip from fingerprint images and use it as shear input. The center of a fingerprint is registered and tracked by group delay spectrum (GDS) which refers to the frequency differential of the phase spectrum of a polar form filter. GDSs of the vertical and horizontal lines crossing the center are calculated and saved as a template. Each row and column line in the fingerprint after deformation is compared with the template and Euclidean norms are calculated. Since the fingerprints change continuously, the Euclidean norms are weighted by the distance of the currently searched line and the previously detected line that passes through the center. The lines with the minimum weighted norms are selected and the coordinate of the center is determined. The vertical and horizontal lines intersecting at the center divided the fingerprint into four quadrants. Eccentricity is calculated by the ratios of areas of the four quadrants. Since eccentricity has linear correlations with sliding distance and shear force, it can be used to determine the direction and magnitude of shear input. A 2D pointing experiment showed that the average time used by the proposed

method is slightly longer that a trackpad. To the best of our knowledge, this is the only study that used the deformation of the finger for shear input. However, there are a few issues associated with this technique to be addressed. Firstly, only 2D analog input can be realized using this technique, which is insufficient for 3D object manipulation tasks. Secondly, eccentricity is unable to distinguish between shear force and finger rotations along lateral and longitudinal axes since both actions will cause the area ratios of four quadrants and eccentricity to change. Thirdly, both GDS tracking and eccentricity calculation accuracy will degrade or even fail when the finger rotates along the axis parallel to the surface normal.

2.3 Fingerprint Deformation

Fingerprint deformation has been widely researched in the field of fingerprint recognition to improve fingerprint matching performance, synthesize realistic fingerprints, and detect spoof fingerprints [1, 8, 14, 47]. Si et al. [47] proposed an algorithm to detect and rectify distortion from a single fingerprint image. A database of deformed fingerprints with the corresponding deformation filed is constructed for reference. In the testing stage, the nearest neighbor of an input fingerprint is found in the database and the corresponding deformation field is used to rectify the input fingerprint. Cui et al. [14] utilized minutiae-based thin-plate-spline (TPS) transformation and phase demodulation for fingerprint registration. The deformation field is created by the unwrapping phase difference between two fingerprints. Cappelli et al. [8] divide a deformed fingerprint into three regions: a close-contact region (center region) where skin slippage is not allowed; an external region (boundary region) where the skin is dragged by finger movement; a transitional region where an elastic distortion is produced. Points in the close-contact region remain fixed while points in the external region move rigidly. A plastic distortion model is then given based on these observations. Antonelli et al. [1] introduced an algorithm to detect fake fingerprints using deformation extracted from adjacent frames in a fingerprint sequence. Optical flows are firstly computed for adjacent frames and the deformation field is then calculated using optical flows. These researches demonstrated the rich information contained in deformed fingerprints. However, the above scenarios and methods differ significantly from fingerprint deformation-based human-computer interaction.

3 PRINTSHEAR: SHEAR INPUT USING FINGERPRINT DEFORMATION

To address the issues associated with previous approaches in the domain of shear input, we propose a shear input technique utilizing fingerprint deformation. Feature points from adjacent frames in the captured fingerprint sequence are first extracted and matched. Lateral, longitudinal, and rotational deformations are then determined based on distance and the relative position between matched points. Deformations are finally mapped to 3DOF input commands and 6DOF expansion can be achieved through recognition of the contact region of the touching finger. The overall workflow is shown in Figure 2.

3.1 Feature Point Extraction and Matching

Minutiae are commonly used for fingerprint recognition due to their uniqueness and biological stability [37]. However, minutiae are sparsely and unevenly distributed on fingers, which leads to inaccurate calculations of deformations, especially for small touching areas. Therefore, we use the speeded-up robust features (SURF) [4] algorithm to extract feature points from fingerprint images. Although the SURF algorithm is widely used to extract feature points for generic image recognition, fingerprint matching performance is generally unsatisfactory using SURF feature points since fingerprints collected at different times may vary significantly under various conditions including touch pressure, skin wetness, etc. Since the difference between adjacent frames in a fingerprint sequence is small, SURF feature points can provide fast and accurate performance in the scenario of human-computer interaction.

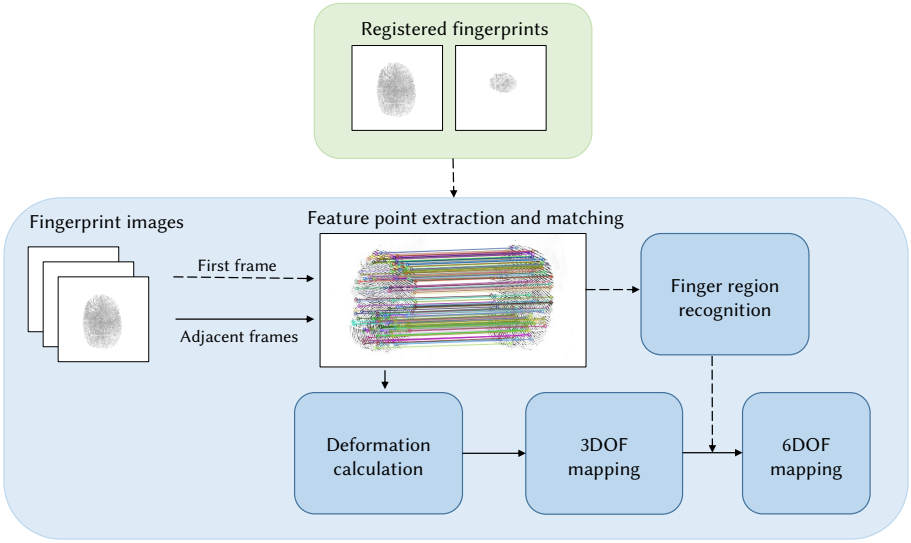


Fig. 2. Overall workflow of PrintShear. Solid arrows indicate the process to create 3DOF control commands using adjacent frames. Dashed arrows indicate the finger region recognition process used for 6DOF control command expansion.

After feature point extraction, we use the fast library for approximate nearest neighbor (FLANN) [40] to find the matching points and filter unreliable matchings using Lowe's ratio test [36]. Finally, random sample consensus (RANSAC) is used to find the optimal transformation matrix and remove outliers.

3.2 Deformation Calculation

Once feature point pairs are found, dispositions of each pair of feature points are calculated. We classify finger actions into three categories: static, push and slide based on the proportions of moving points. Two threshold values are used to remove the jitter effect and improve robustness. From our pilot test, we observed that the false feature point matchings accounted for less than 5% of all point matchings. Therefore, a 5% redundancy on both sides of a push action detection range is enough to prevent the false finger action caused by false feature point matchings. A larger redundancy value can also be used at the cost of slightly reduced accuracy. Finger is considered static if less than 5% of matching points are moving points. Push action is detected when the proportion of moving points is between 5% and 95%. Slide action is detected when the proportion of moving points exceeds 95% (see Figure 3).

Lateral and longitudinal deformations are calculated as the mean displacement of all moving points in the x- and y-directions respectively. Rotational deformation is calculated from the transformation matrix.

3.3 Input Command Mapping

Lateral, longitudinal, and rotational displacement can be mapped to control rotation or translation in the x-, y- and z-directions based on the task requirement. Since 6DOF control is required for the manipulation of a 3D object, we use two regions in the finger to control rotation and translation separately (see Table 1). A user first registers two regions in a finger with SURF feature points and descriptors extracted as templates. In the testing phase, the first frame in the fingerprint image sequence is used as a query image and matched with templates. A matching score is calculated as the ratio of matched points to all extracted points. The template with a higher

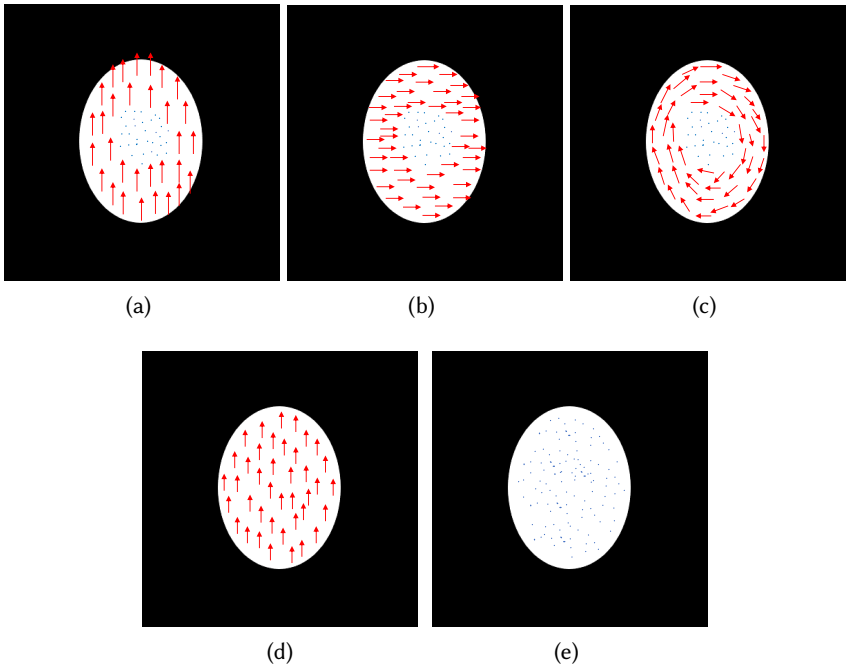


Fig. 3. Different finger actions (red arrows indicate moving point pairs; blue points indicate static points): (a) push up; (b) push right; (c) rotate clockwise; (d) slide up; (e) static.

Table 1. 6DOF control mappings

	X-rotation	Y-rotation	Z-rotation	X-translation	Y-translation	Z-translation
Region 1	Lateral deformation	Longitudinal deformation	Rotational deformation			
Region 2				Lateral deformation	Longitudinal deformation	Rotational deformation

matching score is selected and the following inputs are fixed to the DOFs associated with it until the end of the fingerprint sequence. Another approach to DOF expansion is to use multiple fingers. The number of touching fingers is firstly determined through fingerprint segmentation. Rotations and translations are then controlled by one and two fingers respectively. Fingerprint registration and recognition are not required for this approach, however, a larger fingerprint sensing area is required. Since PrintShear mainly focuses on applications on small sensors, operations using multiple fingers are not utilized in this study.

Input can be classified into three categories according to the device's resistive force. Isotonic devices have zero or constant resistance to measure the movement of the devices (e.g., the computer mouse). Isometric devices have infinite resistance and sense applied force (e.g., TrackPoint). Elastic devices have varying resistance which increases with displacement (e.g., SpaceMouse). Input can be mapped to control the movement of 3D objects via rate or position. Zhai [55] showed that isotonic devices perform better with position control while isometric and

elastic devices perform better with rate control. Finger shear input can be classified as elastic input since the finger skin is elastic and resistance increase with the level of deformation. Therefore, we implement PrintShear with rate control in this study. Inspired by the PRISM [18] technique, we implemented a three-stage dynamic control gain strategy, where the control gain is scaled down for deformations below a threshold value and scaled up for deformations above another threshold value.

4 EVALUATION

We conducted a finger deformation experiment and four 3D object docking tasks to evaluate the feasibility and performance of PrintShear:

- Evaluation of the deformation ability of different fingers and regions.
- 3DOF rotation task using different fingers and regions.
- 2DOF translation task using different input methods.
- 3DOF rotation task using different input methods.
- 6DOF manipulation task using different input methods.

The deformation ability of a finger indicates the feasibility to use it for fingerprint-deformation-based input. Therefore, we designed a finger deformation experiment to find the range of deformation achieved by different fingers and regions. The other four tasks were 3D docking tasks which required the participants to move an object to a target location and/or orientation. We counter-balanced the order of input methods each participant used during each task with a Latin-square approach to reduce the bias from the learning effect. The completion time for each trial was recorded for evaluation. We further analyzed the performance of all participants under the assumption of a fixed effect model. The dependent variable for task 1 was the deformation range while the dependent variable for the other tasks was completion time.

4.1 Participants

This study was approved by the institutional review board (IRB) at Tsinghua University. We recruited 12 participants (8 males, 4 females, all right-handed) aged between 19 and 32 from our university. All participants provided informed consent and their information was kept confidential. No participants had any medical issues which would affect the operation of each input device. No participants had previous experience with SpaceMouse or shear input. All participants had experience using the 3DOF touch input, while only four had a little experience of 6DOF object manipulation. Six participants had experience with TrackPoint. All participants were given unrestricted time to practice with different techniques until they were satisfied with their ability to use each technique. Questionnaires were filled by participants after completing all tasks, including a NASA-TLX (raw TLX) [24] to assess workload and a questionnaire to assess fatigue level and subjective ratings for different techniques. The study adhered to ethical guidelines for research involving human subjects.

4.2 Apparatus

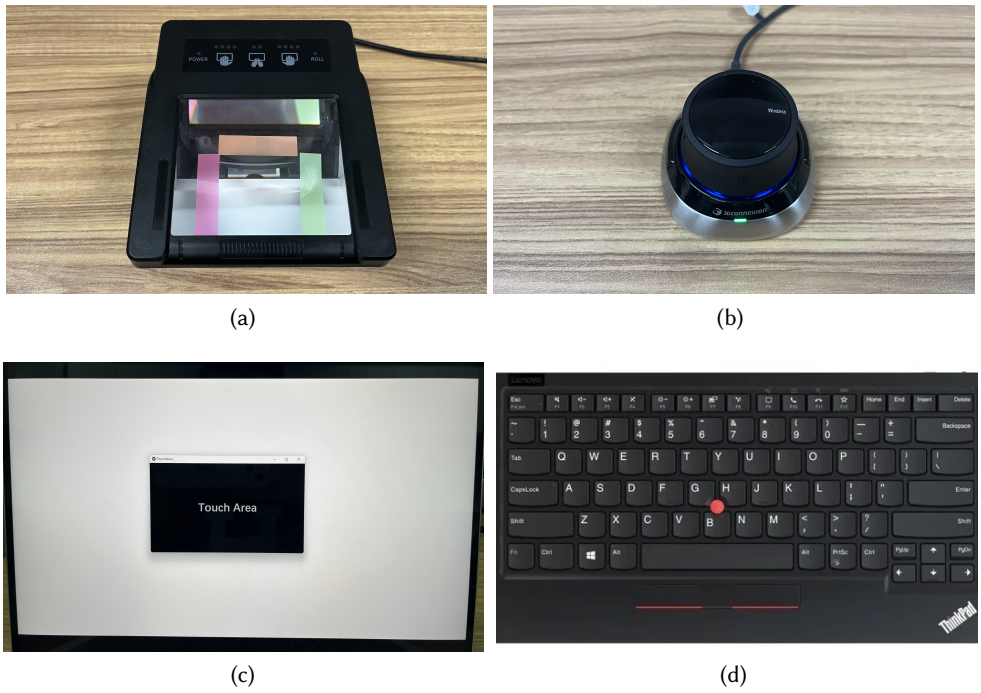
The scene of the docking tasks was displayed on a 24" LED monitor with a resolution of 1920×1080 pixels. Four input methods were used in the evaluation session: conventional touch input with gestures, TrackPoint, SpaceMouse, and PrintShear. A Dotutech DF500[2] optical fingerprint scanner was used to capture real-time fingerprint images with a resolution of 800×750 pixels (500 PPI) at a frame rate of 30Hz. It is worth noting that fingerprint deformation measurement only requires a small area of this fingerprint scanner. Therefore the proposed technique applies to the small fingerprint sensors commonly used in mobile phones. We used this scanner for our experiments because its SDK facilitates the acquisition of higher frame-rate images. Conventional

[2]http://www.dotutech.com/en/pro_d.php?id=3

touch input was performed on the touchscreen of a Huawei Matebook X[3] laptop. The screen was set vertically upright with a small tilt angle so that the user can interact with the screen comfortably. The actual touch area is limited to 7 cm × 12 cm square. Multi-touch inputs were acquired using Kivy[4] framework and the following gesture mappings were used as in [5]:

- One-finger movement: xy-rotation
- Two-finger movement: xy-translation
- Two-finger pinching: z-translation
- Two-finger swiveling: z-rotation

6DOF inputs from the SpaceMouse were acquired using a pyspacenavigator package[5]. We followed the original 6DOF control mappings for SpaceMouse without modification. The TrackPoint from a Lenovo ThinkPad[6] was used in the 2DOF translation task. All input devices are shown in Figure 4.



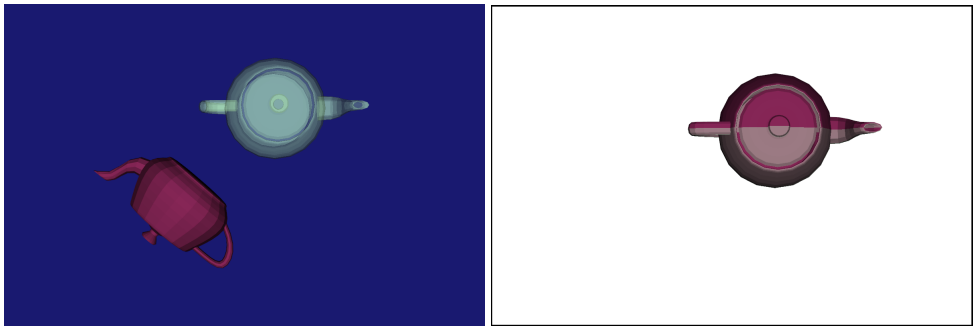
(a) (b) (c) (d)

Fig. 4. Input devices used in this study: (a) fingerprint scanner; (b) SpaceMouse; (c) touchscreen; (d) TrackPoint.

The 3D docking task was programmed using VTK[7] (see Figure 5). We used a 3D teapot model for the docking task due to the simple structure and clear indication of orientation. The object for manipulation was in red while the target object was in green with 50% opacity. The scene had a blue background with a fixed viewpoint. The

[3]https://consumer.huawei.com/en/laptops/matebook-x-2020/
[4]https://kivy.org/
[5]https://github.com/johnhw/pyspacenavigator
[6]https://www.lenovo.com/us/en/c/laptops/thinkpad
[7]https://vtk.org/

depth of the object for manipulation could be perceived based on the relative position and size of the object. We randomly generated 10 target orientations and 20 target positions beforehand. The target positions were divided into two groups based on the distance from the object. The distance between the first group of positions and the target is less than 80 pixels on each axis, and the distance between the second set of positions and the target is between 80 and 300 pixels on each axis. Statistics for the target position groups are shown in Table 1. During each docking task, input methods were counterbalanced to avoid bias from learning effects. A timer started when a participant makes the first movement using each input device. The background color turned white when the angle difference in each axis was less than 2 degrees and the distance from the object to the target position was less than 2 pixels. The timer stopped when the object stayed at the target position and orientation for 0.5 seconds. To facilitate a fair comparison, the refresh rate was set to 30 Hz for all devices.



(a) (b)

Fig. 5. 3D object docking task scene: (a) initial scene; (b) completion scene.

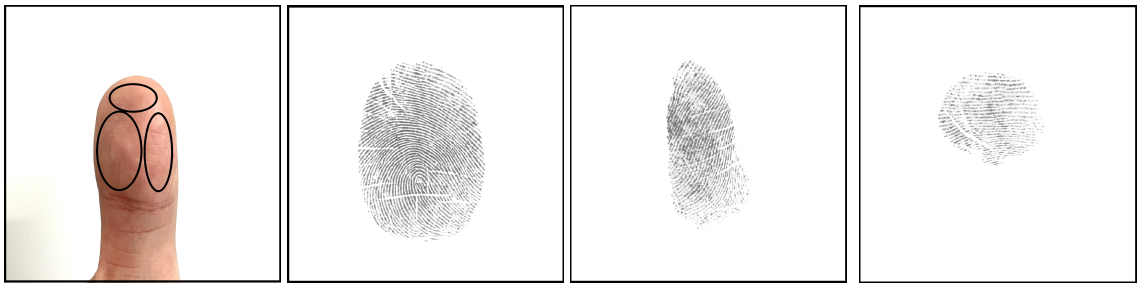
Table 2. Statistics of target orientations and position of randomly generate position groups

	X-axis		Y-axis		Z-axis	
	Mean	STD	Mean	STD	Mean	STD
Orientation group (in degrees)	17.48	98.53	-35.58	90.02	36.46	99.31
Position group 1 (in pixels)	43.88	21.89	32.09	20.00	48.10	22.06
Position group 2 (in pixels)	164.56	82.11	120.34	75.01	180.38	82.76

4.3 Tasks

4.3.1 Task 1: Evaluation of the deformation ability of different fingers and regions. During this experiment, participants were asked to press a finger on the fingerprint scanner while pushing towards four directions (up, down, left and right) and rotate in two directions (clockwise and anticlockwise) until sliding occurred. Deformations between adjacent frames were calculated using the algorithm in section 3 and accumulated to a deformation range for each axis. The fingers used in this experiment were thumbs and index fingers, while the

regions used were top, center and side of each finger. Different regions of a finger are shown in Figure 6. Each direction was repeated twice, resulting in 12 (participants) × 2 (repetitions) × 6 (directions) × 2 (fingers) × 3 (regions) = 864 trials.



(a) (b) (c) (d)

Fig. 6. Finger regions used in this study: (a) different regions of the finger (right thumb); (b) fingerprint of center region; (c) fingerprint of side region; (d) fingerprint of top region.

4.3.2 Task 2: 3DOF rotation task using different fingers and regions. This task aims to evaluate the performance of different fingers and regions in 3DOF control. During this task, participants were asked to rotate the object to a target orientation. Our pilot study found that the separation of actions in the x-, y- and z-axis led to increased clutching and operation time. Therefore, DOF separation was not applied in this task and participants were able to make adjustments in all directions without clutching. Again, participants were asked to use the top, center, and side regions of their thumbs and index fingers to complete this task, resulting in 12 (participants) × 10 (repetitions) × 2 (fingers) × 3 (regions) = 720 trials.

4.3.3 Task 3: 2DOF translation task using different input methods. Movements in 2D are a common task in various scenarios, including cursor movement, drawing, map navigation, etc. In this task, we compared the performance of PrintShear with an isometric TrackPoint and conventional touch input in 2DOF control. To further investigate the relationship between performance and moving distance, we used two randomly generated groups of target positions. A total of 12 (participants) × 10 (repetitions) × 3 (input methods) × 2 (distance groups) = 720 trials were completed in this task.

4.3.4 Task 4: 3DOF rotation task using different input methods. This task compares the 3DOF control performance of PrintShear with SpaceMouse and conventional gesture-based touch input. The same setup was used as task 2, except for the input methods. Participants were allowed to use their preferred region and finger for PrintShear, and their selections were recorded. Position control was used for the isotonic conventional touch input while rate control was used for the elastic SpaceMouse and PrintShear. A total of 12 (participants) × 10 (repetitions) × 3 (input methods) = 360 trials were completed in this task.

4.3.5 Task 5: 6DOF manipulation task using different input methods. Task 5 investigates the performances of PrintShear, conventional touch input, and SpaceMouse in a realistic 6DOF 3D object manipulation scenario. Before using PrintShear, participants were asked to register two regions of a preferred finger to provide control for rotation and translation separately. To eliminate the time for switching fingers, only one finger was allowed to be used by each participant. The main purpose of this task is to evaluate the potential of PrintShear in the scenario of single-hand operation on a limited fingerprint sensing area (e.g. mobile phones). Therefore, using one

finger from each hand to control translation and rotation separately at the same time is not evaluated in this task, even though it has the potential to further improve the performance of PrintShear. Similar to task 3, two groups of target positions were used for this task. A total of 12 (participants) × 10 (repetitions) × 3 (input methods) × 2 (distance groups) = 720 trials were completed in this task.

4.4 Performance Analysis

We used the Shapiro-Wilk test to check the normality of data in all tasks and the subjective ratings. Results showed that only the deformation ranges and the subjective ratings followed a normal distribution. Therefore, we plotted the means and 95% confidence intervals (CI) for the deformation ranges and the subjective ratings. Paired t-tests with Benjamini-Hochberg corrections were performed to verify the significance. Since most of the completion times in tasks 2-5 failed the normality check, we used violin plots to reflect the real distribution of original data and paired performance differences. Non-parametric Wilcoxon signed-rank tests [50] with Benjamini-Hochberg corrections were used to verify the significance of completion time differences between different input methods.

4.4.1 Task 1: Evaluation of the deformation ability of different fingers and regions. The deformation ability of different regions and fingers was evaluated using fingerprint deformation ranges. The deformation ranges for each finger and region in all three directions are shown in Figure 7. When the data is grouped by fingers, the thumb showed a better overall deformation ability than the index finger ($p < 0.05$). When the data are grouped by regions, the center region showed the best overall deformation ability than other regions ($p < 0.001$) and the side region showed a better deformation ability than the top region ($p < 0.001$). This could be explained by the differences in touching area sizes, where a larger area has more potential to deform. In addition, all fingers and regions showed better deformation abilities in the lateral directions ($p < 0.001$). The observations on deformation abilities provide a basis for finger and region selection as well as sensitivity mapping.

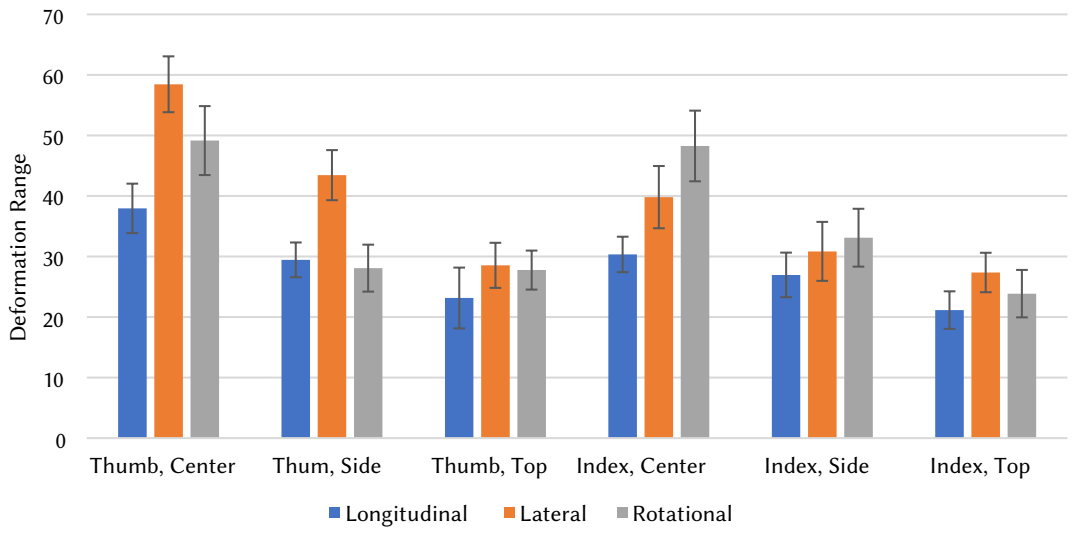


Fig. 7. Lateral, longitudinal (in pixels) and rotational (in degrees) deformation ranges for different regions and fingers. Error bars: 95% CI.

4.4.2 Task 2: 3DOF rotation task using different fingers and regions. Completion times for the 3DOF rotation task using different fingers and regions are shown in Figure 8. The average completion time ranged from 7.41 s (index finger, center region) to 9.72 s (thumb, center region). Surprisingly, the center region of the thumb performed worst among all regions of both fingers. During the evaluation, we observed frequent collisions between the hands of participants and the bottom edge of the fingerprint scanner when the participants made rotations using the center of their thumbs. This affected the actual deformation range and therefore increased clutch time. The performance is expected to improve when using a fingerprint-sensing device with a smaller form factor, such as the fingerprint sensor in mobile phones. In addition, when the completion times were grouped by fingers (thumb and index finger), no significant performance difference was observed between the thumb and the index finger ($p = 0.53$).

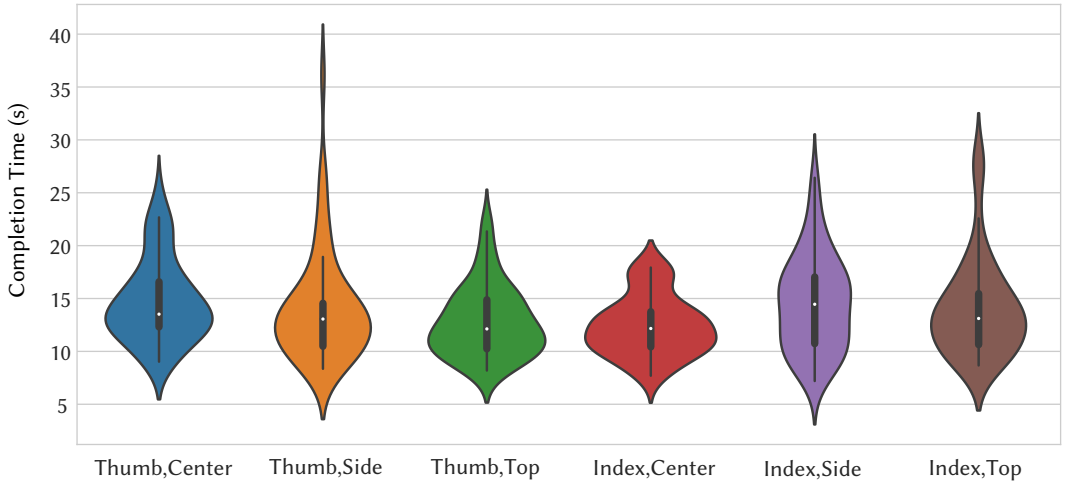


Fig. 8. Completion times (in seconds) for 3DOF rotation task using different regions and fingers.

4.4.3 Task 3: 2DOF translation task using different input methods. We observed that one participant chose to use the top region of the thumb to operate PrintShear, while others chose to use the center region of the index finger. The performance results are depicted in Figure 9 and the completion time differences are depicted in Figure 10 In the short-distance scenario, conventional touch input had the shortest completion time (2.58 s). The performance was significantly better than PrintShear (4.45 s, $p < 0.001$) and TrackPoint (4.20 s, $p < 0.001$). In the long-distance scenario, conventional touch input also had the shortest completion time (5.13 s) and the differences were significant compared to PrintShear (6.73 s, $p < 0.01$) and TrackPoint (6.32 s, $p < 0.01$). However, the percentage of the time difference between touch input and other methods decreased from 41.97% (PrintShear) and 38.46% (TrackPoint) in the short-distance scenario to 23.80% (PrintShear) and 18.79% (TrackPoint) in the long-distance scenario. This could be explained by the isotonic nature of conventional input. Isotonic input devices with position control enable precise movements, but the performance degrades due to significant clutching in long-distance movements [10]. In addition, no significant differences were observed between TrackPoint and PrintShear in both short-distance ($p = 0.57$) and long-distance ($p = 0.39$) scenarios.

4.4.4 Task 4: 3DOF rotation task using different input methods. We observed that one participant chose to use the top region of the thumb to operate PrintShear, while others chose to use the center region of the index finger.

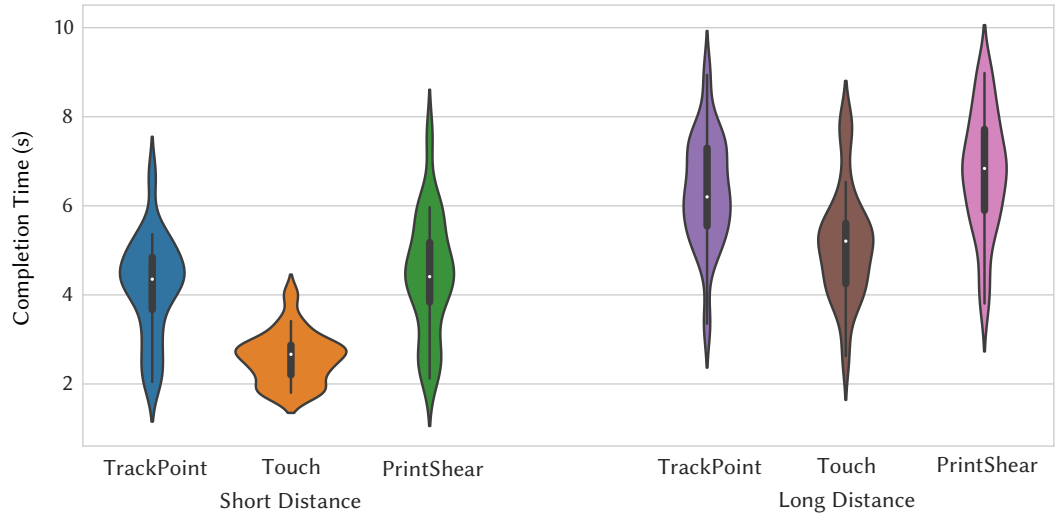


Fig. 9. Completion times (in seconds) for 2DOF translation task using TrackPoint, conventional touch input and PrintShear.

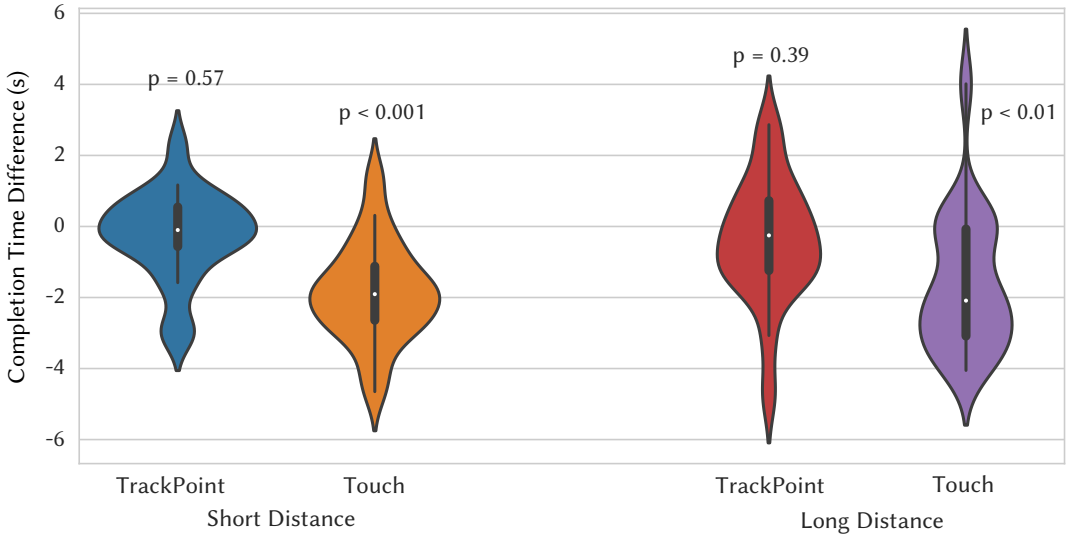


Fig. 10. Completion time differences (in seconds) for the 2DOF translation task using TrackPoint and conventional touch input compared to PrintShear.

The performance results of 3DOF rotation task is shown in Figure 11. The completion time using each input method was 7.15 s for PrintShear, 7.76 s for SpaceMouse, and 8.20 s for conventional touch input. PrintShear showed significant improvements compared with SpaceMouse ($p < 0.05$) and conventional touch input ($p < 0.05$). PrintShear achieved a 14.65% reduction in completion time compared with conventional touch input. During this task, we observed that participants spent a lot of time on two-finger swiveling while using conventional touch

input. Since the angle range for the swiveling gesture is relatively small, more clutching actions were performed when the magnitude of the angle difference in the z-axis is large. We also observed that the controlled object continued to move a short distance when the SpaceMouse was released since the device requires a short time to return to the neutral position. In contrast, the object stopped immediately when the participant lifted the finger from the fingerprint scanner when using PrintShear.

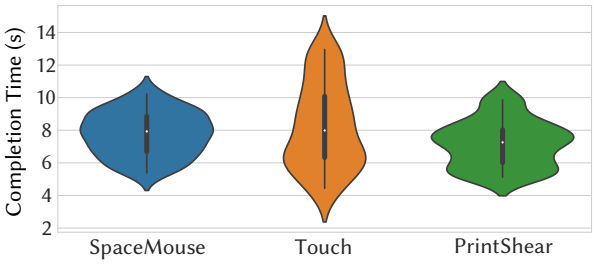


Fig. 11. Completion times (in seconds) for 3DOF rotation task using SpaceMouse, conventional touch input and PrintShear.

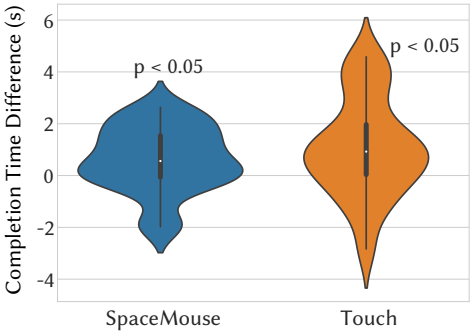


Fig. 12. Completion time differences (in seconds) for the 3DOF rotation task using SpaceMouse and conventional touch input compared to PrintShear.

4.4.5 Task 5: 6DOF manipulation task using different input methods. We observed that all participants chose to use the top and center region of the index finger to operate PrintShear. The participant who chose to use the top region of the thumb in previous tasks decided to use the index finger in this task. The participant commented that the index finger is easier to use when more than one region needs to be used. The performance results of the 6 DOF manipulation tasks are shown in Figure 13. In the short-distance scenario, the completion time for SpaceMouse, conventional touch input, and PrintShear were 28.03 s, 29.22 s, and 24.02 s respectively. PrintShear showed the best performance and the difference was significant compared with conventional touch input ($p < 0.05$), but not with SpaceMouse ($p = 0.26$). In the long-distance scenario, the completion time for SpaceMouse, conventional touch input, and PrintShear were 40.13 s, 45.11 s, and 35.29 s respectively. PrintShear showed the best performance and the difference was significant compared with conventional touch input ($p < 0.05$), but not with SpaceMouse ($p = 0.11$). On average, the participants spent 17.82% and 21.76% (for an overall average of 19.79%) less time to complete each task using PrintShear than using conventional touch input.

4.4.6 Subjective ratings. Based on the responses in the questionnaires, we depicted the subjective ratings in Figure 15 and Figure 16. The workload results using NASA-TLX (see Figure 15) showed that the least overall workload was required by PrintShear. It should be noted that the physical demand for PrintShear was the highest among all methods. This is reasonable since deformation is created by shear force which requires a relatively large pressing force. It is noteworthy that PrintShear had the lowest frustration level. Thanks to the separation of translation and rotation, participants rarely encountered misoperations. In contrast, xy-translation and xy-rotation actions were frequently mixed for SpaceMouse while z-translation and z-rotation actions were frequently mixed for conventional touch input. Fatigue level and user preference score are shown in Figure 16. Similar to the physical workload in Figure 15, PrintShear had the highest fatigue level. SpaceMouse had the highest user preference score due to the balance between performance and fatigue level. PrintShear had a slightly lower user preference score than SpaceMouse. Points were deducted by participants due to their fatigue level

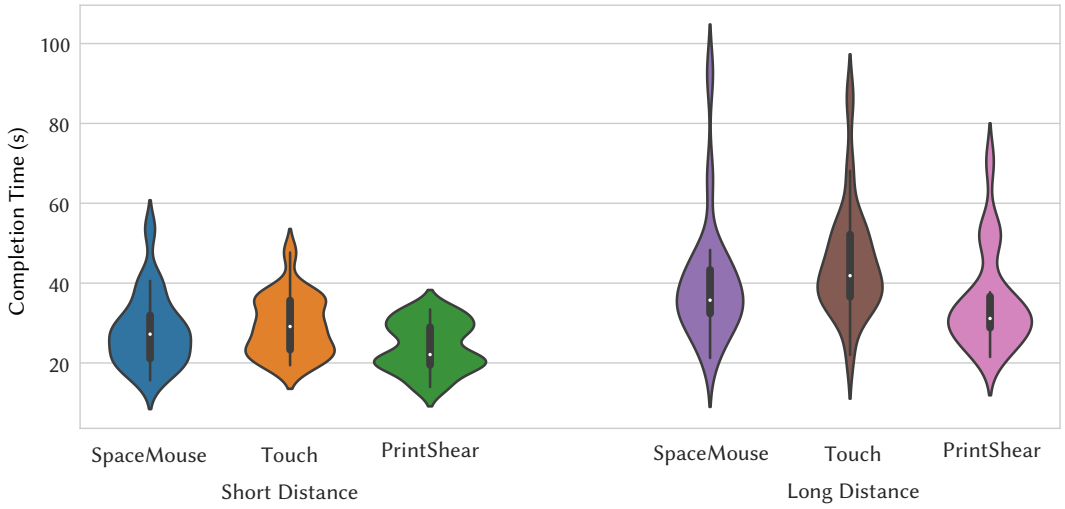


Fig. 13. Completion times (in seconds) for 6DOF manipulation task using SpaceMouse, conventional touch input and PrintShear.

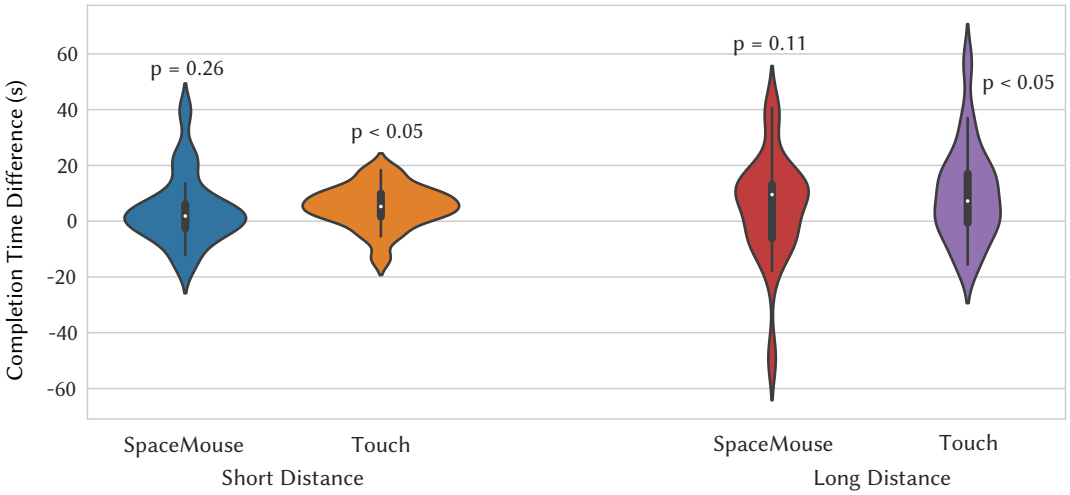


Fig. 14. Completion time differences (in seconds) for the 6DOF manipulation task using SpaceMouse and conventional touch input compared to PrintShear.

after using PrintShear. conventional touch input had the lowest user preference score. The primary negative feedbacks from the participants are high physical workload for swiveling and frequent gesture confusion.

5 LIMITATIONS AND FUTURE WORK

Although experiment results demonstrated the feasibility and performance of fingerprint deformation-based shear input. This study has the following limitations:

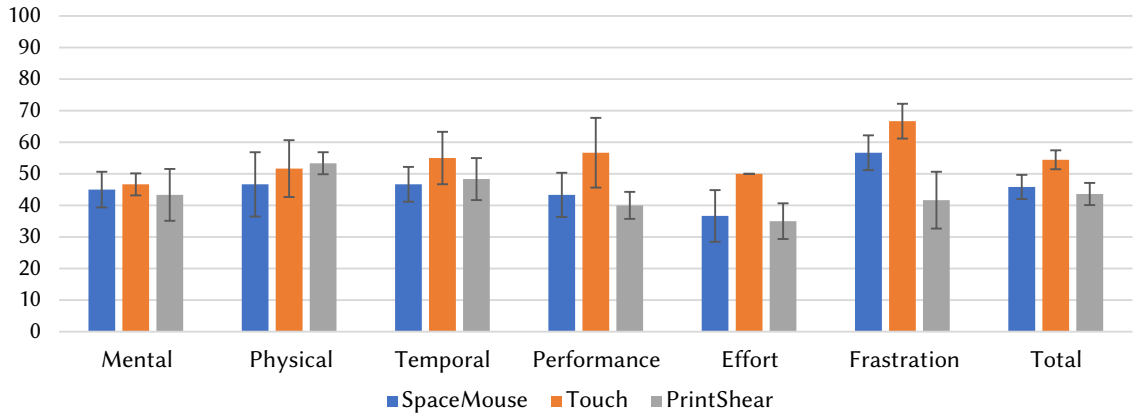


Fig. 15. Workload measurement in NASA-TLX units. Error bars: 95% CI.

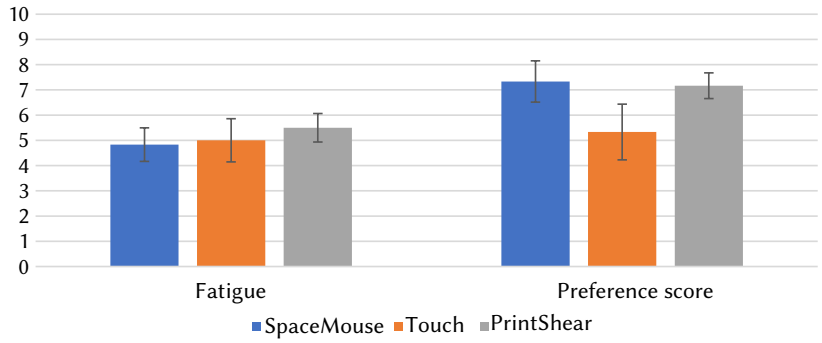


Fig. 16. Fatigue level and user preference score. Error bars: 95% CI.

Fingerprint images were captured using a large standalone optical fingerprint scanner instead of under-screen fingerprint sensors in mobile devices or fingerprint sensors in laptops. This study focused on demonstrating the feasibility of fingerprint deformation-based shear input. Further studies should be conducted to explore the performance of PrintShear on mobile devices and laptops. Since the fingerprint sensors of these computing devices are not open to third-party software due to security concerns, such a study requires the cooperation of mobile phone and laptop manufacturers. In addition, our pilot study showed that a fingerprint sensing area with a size slightly larger than the center region of the thumb (3 cm × 4 cm, see Fig 6b) is enough for PrintShear to operate on since large movement of the finger is not required. We further tested a smaller sensing area (2 cm × 2 cm) which only captured a partial fingerprint. The accuracy of the operation was reduced since the deformation of fingerprints tended to happen in the boundary region and fewer moving points were detected in the partial fingerprint. We plan to comprehensively investigate deformation-based input using smaller sensors in our future studies and aim to solve it with better algorithms. Moreover, the device used in this study had a low refresh rate, which affected manipulation performance. To measure finer deformations and reduce time delay, higher frame rates are required.

Only 12 participants were recruited to conduct the experiments in this study. One of the major focuses of our future studies is to systematically investigate the feasibility of PrintShear through a comprehensive user study with more coverage of age groups, finger skin conditions, occupations, and medical conditions, and further improve the shear force measurement algorithm based on the discovered problems.

The proposed method uses fingerprint matching to achieve 6DOF input, which requires fingerprint registration beforehand. The fingerprint recognition performance will be affected by poor finger conditions. In addition, a user can only operate the device with the registered finger and regions. This feature can be useful since it adds a level of security for applications where continuous authentication is required [17]. Most user authentication systems used one-time authentication such as personal identification numbers (PIN), passwords, and fingerprint scans. It leaves the device unprotected once the initial access has been granted. Various continuous authentication systems have been proposed to solve this issue utilizing touch dynamics, face recognition, etc. [17, 43, 48]. Since PrintShear continuously collects fingerprints, continuous authentication can be achieved without using additional sensors. Moreover, fingerprint deformation can also be used to detect fingerprint spoofs presented to the authentication system including fabricated gummy fingers and printed fingerprints [37]. However, the collection of fingerprints will bring privacy issues regarding possible exposure of biometric information. PrintShear uses the the SURF algorithm for feature point extraction which is commonly used for generic image alignment. The extracted feature points and their descriptors are different from standard fingerprint minutiae templates, used in most fingerprint recognition systems. When fingerprint images are captured by the system, only the feature points and descriptors are saved, further reducing the risk of biometric information exposure. Both 6DOF control and fingerprint verification require a registration step, where templates of fingerprints are created and stored in the system. To protect the templates, cryptographic techniques should be integrated into the system [45]. Other countermeasures such as a trusted execution environment, which is isolated from the normal processing environment can also be used to store sensitive biometric information [16].

In this study, we only implemented and evaluated one possible finger and region combination where participants controlled translations and rotations separately using two regions of the same finger. A similar design to the gamepad can be used for PrintShear where two thumbs are used to control translations and rotations at the same time. Users are unable to rotate their thumbs while holding and operating the mobile phone with one hand. More regions could be explored to provide the missing DOFs and fulfill 6DOF control in this situation, which is unable to be realized using conventional touch input.

We evaluated the deformation abilities of different regions and fingers. However, the results have not been used for control gain adjustments. Research in [9] showed that similar controllability can be achieved for elastic input devices with different stiffness levels when control gain is adjusted based on stiffness. Therefore, performances could be improved for the fingerprint regions with smaller deformation abilities.

Only push actions (partial sliding) are utilized by PrintShear, considering the main application scenario is on small fingerprint sensors in mobile devices. Sliding actions could be used to provide additional input modalities to further improve the performance of PrintShear. The study in [10] introduced an approach that combined position and rate control to reduce clutching. A similar approach can be used by PrintShear where position control is used for push actions while rate control is used for sliding actions. Apart from the dynamic control gain strategy adopted in this study, other approaches could be used to further improve the performance of PrintShear. For example, Barrett et al. [2] introduced a negative inertia technique to reduce sluggishness and improve the performance of isometric input devices.

We only compared TrackPoint, SpaceMouse, and conventional touch input using docking tasks in this study. Other promising input modalities such as finger pose, joysticks, and keyboard+mouse are not compared. PrintShear can be used for many applications, such as video games, sketching, and menu navigation, as illustrated in [22, 28, 30, 41, 51]. However, only 3D object manipulation was evaluated in this study since it includes basic

operations required for other interaction tasks. The performance of an input device in a 3D object manipulation task reflects the performance in other applications.

Lastly, deformation-based input requires a more physical workload to complete. Similar observations are reported in other research on shear force-based interactions [30]. The degree of deformation is proportional to the applied force. One possible solution to solve the fatigue issue is to increase the control gain (sensitivity) of the command mapping. Then less force (deformation) is required to perform an action at the cost of reduced accuracy. In addition, since fingerprint deformation is created by friction, a transparent layer with a higher friction coefficient would help to create the same amount of friction with less pressing force. A major focus of our future research is to explore a more relaxed way to perform and map shear input.

6 CONCLUSION

Although fingerprint sensors have been widely implanted in laptops, mobile phones, and tablets, their use is limited to user authentication. Research that considers fingerprint sensors as a more versatile input device is emerging., but there is still much room for exploration. In this paper, we proposed PrintShear, a shear input method based on fingerprint deformation. Lateral, longitudinal and rotational deformations can be used for 3DOF control and further expansion of DOFs can be achieved through recognition of the contact region of the touching finger. We evaluated the deformation ability of different regions in different fingers and demonstrated the feasibility of PrintShear for each region. Four docking tasks in different scenarios were conducted to evaluate the performance of PrintShear. Experiment results showed that PrintShear outperformed conventional touch input in 3DOF and 6DOF object manipulation tasks with 14.65% and 19.79% reduction in completion time respectively. Further comparisons with TrackPoint and SpaceMouse demonstrated the competitiveness compared to these traditional computer input devices. A unique advantage of PrintShear is that even higher DOFs can be realized by using multiple fingers, thanks to the fingerprint recognition capability. Considering the increasingly embedding of fingerprint sensors in various computing devices, utilizing its shear input capability can be a valuable by-product.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for providing many helpful suggestions that improved the paper and the participants for their participation in our studies. This work was supported in part by the National Natural Science Foundation of China under Grant 61976121.

REFERENCES

[1] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni. 2006. Fake Finger Detection by Skin Distortion Analysis. *IEEE Transactions on Information Forensics and Security* 1, 3 (2006), 360–373. https://doi.org/10.1109/TIFS.2006.879289

[2] Robert C Barrett, Edwin J Selker, Joseph D Rutledge, and Robert S Olyha. 1995. Negative Inertia: A Dynamic Pointing Function. In *Proceedings of the Conference Companion on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 316–317.

[3] Anil Ufuk Batmaz, Mohammad Rajabi Seraji, Johanna Kneifel, and Wolfgang Stuerzlinger. 2021. No Jitter Please: Effects of Rotational and Positional Jitter on 3D Mid-Air Interaction. In *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2*, Kohei Arai, Supriya Kapoor, and Rahul Bhatia (Eds.). Springer International Publishing, Cham, 792–808. https://doi.org/10.1007/978-3-030-63089-8_52

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

[5] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017. Mouse, Tactile, and Tangible Input for 3D Manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4727–4740. https://doi.org/10.1145/3025453.3025863

[6] 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

[7] 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* (San Francisco, California, USA) *(MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/2371574.2371582

[8] R. Cappelli, D. Maio, and D. Maltoni. 2001. Modelling Plastic Distortion in Fingerprint Images. In *Advances in Pattern Recognition — ICAPR 2001*, Sameer Singh, Nabeel Murshed, and Walter Kropatsch (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 371–378.

[9] Géry Casiez and Daniel Vogel. 2008. The Effect of Spring Stiffness and Control Gain with an Elastic Rate Control Pointing Device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1709–1718. https://doi.org/10.1145/1357054.1357321

[10] Géry Casiez, Daniel Vogel, Qing Pan, and Christophe Chaillou. 2007. RubberEdge: Reducing Clutching by Combining Position and Rate Control with Elastic Feedback. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, Rhode Island, USA) *(UIST '07)*. Association for Computing Machinery, New York, NY, USA, 129–138. https://doi.org/10.1145/1294211.1294234

[11] Ashley Colley and Jonna Häkkilä. 2014. Exploring Finger Specific Touch Screen Interaction for Mobile Phone User Interfaces. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design* (Sydney, New South Wales, Australia) *(OzCHI '14)*. Association for Computing Machinery, New York, NY, USA, 539–548. https://doi.org/10.1145/2686612.2686699

[12] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300442

[13] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300442

[14] Zhe Cui, Jianjiang Feng, Shihao Li, Jiwen Lu, and Jie Zhou. 2018. 2-D Phase Demodulation for Deformable Fingerprint Registration. *IEEE Transactions on Information Forensics and Security* 13, 12 (2018), 3153–3165. https://doi.org/10.1109/TIFS.2018.2841849

[15] 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* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 459–469. https://doi.org/10.1145/3490099.3511123

[16] Jan-Erik Ekberg, Kari Kostiainen, and Nadarajah Asokan. 2013. Trusted Execution Environments on Mobile Devices. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (Berlin, Germany) *(CCS '13)*. Association for Computing Machinery, New York, NY, USA, 1497–1498. https://doi.org/10.1145/2508859.2516758

[17] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security* 8, 1 (2013), 136–148. https://doi.org/10.1109/TIFS.2012.2225048

[18] Scott Frees, G. Drew Kessler, and Edwin Kay. 2007. PRISM Interaction for Enhancing Control in Immersive Virtual Environments. *ACM Trans. Comput.-Hum. Interact.* 14, 1 (May 2007), 2–es. https://doi.org/10.1145/1229855.1229857

[19] 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. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3293–3315. https://doi.org/10.1145/3025453.3025808

[20] Mark Hancock, Thomas ten Cate, and Sheelagh Carpendale. 2009. Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (Banff, Alberta, Canada) *(ITS '09)*. Association for Computing Machinery, New York, NY, USA, 133–140. https://doi.org/10.1145/1731903.1731930

[21] M.S. Hancock, F.D. Vernier, D. Wigdor, S. Carpendale, and Chia Shen. 2006. Rotation and Translation Mechanisms for Tabletop Interaction. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)*. 8 pp.–. https://doi.org/10.1109/TABLETOP.2006.26

[22] 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* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 3149–3152. https://doi.org/10.1145/2207676.2208730

[23] 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

[24] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (2006), 904–908. https://doi.org/10.1177/154193120605000909

[25] 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 (Mar 2022), 22 pages. https://doi.org/10.1145/3517243

[26] Seongkook Heo and Geehyuk Lee. 2011. Force Gestures: Augmented Touch Screen Gestures Using Normal and Tangential Force. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 1909–1914. https://doi.org/10.1145/1979742.1979895

[27] Seongkook Heo and Geehyuk Lee. 2012. ForceDrag: Using Pressure as a Touch Input Modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference* (Melbourne, Australia) *(OzCHI '12)*. Association for Computing Machinery, New York, NY, USA, 204–207. https://doi.org/10.1145/2414536.2414572

[28] Seongkook Heo and Geehyuk Lee. 2013. Indirect Shear Force Estimation for Multi-Point Shear Force Operations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 281–284. https://doi.org/10.1145/2470654.2470693

[29] Da-Yuan Huang, Ming-Chang Tsai, Ying-Chao Tung, Min-Lun Tsai, Yen-Ting Yeh, Liwei Chan, Yi-Ping Hung, and Mike Y. Chen. 2014. TouchSense: Expanding Touchscreen Input Vocabulary Using Different Areas of Users' Finger Pads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 189–192. https://doi.org/10.1145/2556288.2557258

[30] Mengting Huang, Kazuyuki Fujita, Kazuki Takashima, Taichi Tsuchida, Hiroyuki Manabe, and Yoshifumi Kitamura. 2019. ShearSheet: Low-Cost Shear Force Input with Elastic Feedback for Augmenting Touch Interaction. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (Daejeon, Republic of Korea) *(ISS '19)*. Association for Computing Machinery, New York, NY, USA, 77–87. https://doi.org/10.1145/3343055.3359717

[31] 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

[32] Y. Kurita, A. Ikeda, Jun Ueda, and T. Ogasawara. 2005. A Fingerprint Pointing Device Utilizing the Deformation of the Fingertip During the Incipient Slip. *IEEE Transactions on Robotics* 21, 5 (2005), 801–811. https://doi.org/10.1109/TRO.2005.847568

[33] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173934

[34] Bhoram Lee, Hyunjeong Lee, Soo-Chul Lim, Hyungkew Lee, Seungju Han, and Joonah Park. 2012. Evaluation of Human Tangential Force Input Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 3121–3130. https://doi.org/10.1145/2207676.2208727

[35] Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. 2012. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. *Computer Graphics Forum* 31, 7 (2012), 2047–2055. https://doi.org/10.1111/j.1467-8659.2012.03197.x

[36] D.G. Lowe. 1999. Object Recognition from Local Scale-invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2. 1150–1157 vol.2. https://doi.org/10.1109/ICCV.1999.790410

[37] Davide Maltoni, Dario Maio, Anil K. Jain, and Jianjiang Feng. 2022. *Handbook of Fingerprint Recognition, 3rd Edition.* Springer.

[38] Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2012. Integrality and Separability of Multitouch Interaction Techniques in 3D Manipulation Tasks. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (2012), 369–380. https://doi.org/10.1109/TVCG.2011.129

[39] 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* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3132272.3134130

[40] Marius Muja and David G Lowe. 2009. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.. In *Proceedings of International Conference on Computer Vision Theory and Application*. 331–340.

[41] Yuriko Nakai, Shinya Kudo, Ryuta Okazaki, Hiroyuki Kajimoto, and Hidenori Kuribayashi. 2014. Detection of Tangential Force for a Touch Panel Using Shear Deformation of the Gel. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI EA '14)*. Association for Computing Machinery, New York, NY, USA, 2353–2358. https://doi.org/10.1145/2559206.2581149

[42] 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* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4238–4249. https://doi.org/10.1145/2858036.2858179

[43] Vishal M. Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. 2016. Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine* 33, 4 (2016), 49–61. https://doi.org/10.1109/MSP.2016.2555335

[44] 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

[45] Salil. Prabhakar, Sharath. Pankanti, and Anil K. Jain. 2003. Biometric recognition: security and privacy concerns. *IEEE Security & Privacy* 1, 2 (2003), 33–42. https://doi.org/10.1109/MSECP.2003.1193209

[46] Joseph D. Rutledge and Ted Selker. 1990. Force-to-Motion Functions for Pointing. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction (INTERACT '90)*. North-Holland Publishing Co., NLD, 701–706. https://dl.acm.org/doi/10.5555/647402.725310

[47] Xuanbin Si, Jianjiang Feng, Jie Zhou, and Yuxuan Luo. 2015. Detection and Rectification of Distorted Fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 3 (2015), 555–568. https://doi.org/10.1109/TPAMI.2014.2345403

[48] Zdeňka Sitová, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S. Balagani. 2016. HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. *IEEE Transactions on Information Forensics and Security* 11, 5 (2016), 877–892. https://doi.org/10.1109/TIFS.2015.2506542

[49] 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* (Victoria, BC, Canada) *(UIST '09)*. Association for Computing Machinery, New York, NY, USA, 23–32. https://doi.org/10.1145/1622176.1622182

[50] Frank Wilcoxon. 1992. *Individual Comparisons by Ranking Methods*. Springer New York, New York, NY, 196–202. https://doi.org/10.1007/978-1-4612-4380-9_16

[51] Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 193–196. https://doi.org/10.1145/2556288.2557017

[52] 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* (Madeira, Portugal) *(ITS '15)*. Association for Computing Machinery, New York, NY, USA, 47–50. https://doi.org/10.1145/2817721.2817737

[53] 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

[54] Vadim Zaliva. 2012. 3D Finger Posture Detection and Gesture Recognition on Touch Surfaces. In *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. 359–364. https://doi.org/10.1109/ICARCV.2012.6485185

[55] Shumin Zhai. 1996. *Human Performance in Six Degree of Freedom Input Control*. University of Toronto.