RESEARCH-ARTICLE

Touchscreen-based Hand Tracking for Remote Whiteboard Interaction

XINSHUANG LIU, Microsoft Research Asia, Beijing, China

YIZHONG ZHANG, Microsoft Research Asia, Beijing, China

XIN TONG, Microsoft Research Asia, Beijing, China

Open Access Support provided by:

Microsoft Research Asia

Published: 13 October 2024

Citation in BibTeX format

UIST '24: The 37th Annual ACM Symposium on User Interface Software and Technology
October 13 - 16, 2024
PA, Pittsburgh, USA

Touchscreen-based Hand Tracking for Remote Whiteboard Interaction

Xinshuang Liu[*][†]
xinshuangliu@outlook.com
University of California San Diego
USA

Yizhong Zhang[‡]
yizzhan@microsoft.com
Microsoft Research Asia
China

Xin Tong
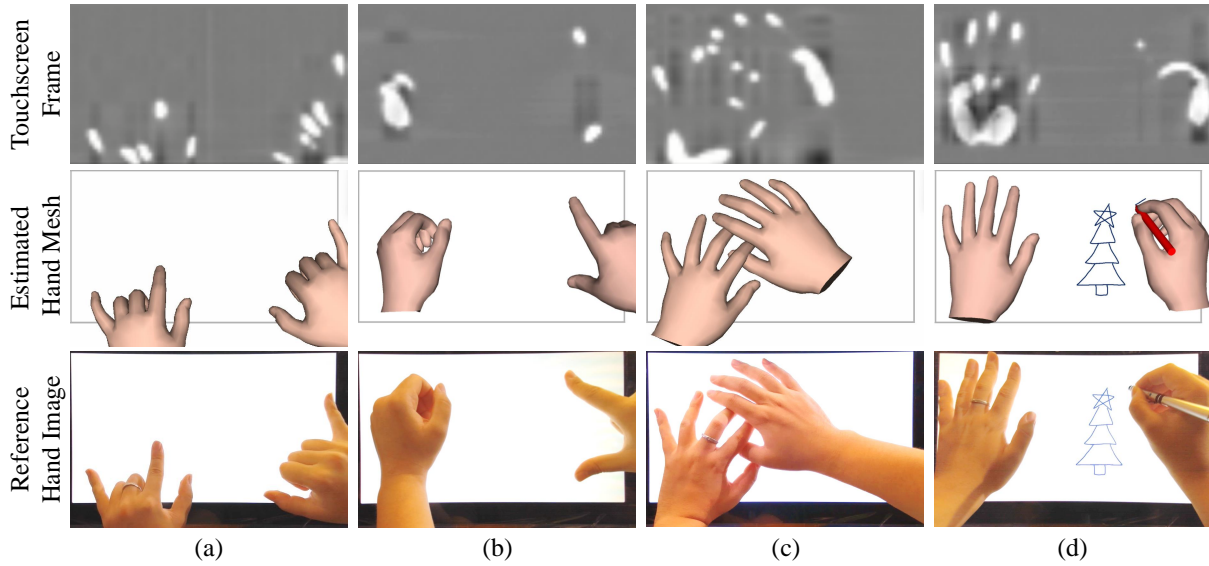xtong.gfx@gmail.com
Microsoft Research Asia
China

Figure 1: Given the capacitive frames of two hands of a subject interacting with the touchscreen (first row), our method reconstructs the corresponding 3D hand poses (second row) in real time. The ground truth hand poses of the captured capacitive frames are displayed in the third row for comparison. Note that our method can accurately reconstruct various poses of two hands, even for overlapped hands as demonstrated in (c).

ABSTRACT

In whiteboard-based remote communication, the seamless integration of drawn content and hand-screen interactions is essential for an immersive user experience. Previous methods either require bulky device setups for capturing hand gestures or fail to accurately track the hand poses from capacitive images. In this paper, we present a real-time method for precise tracking 3D poses of both hands from capacitive video frames. To this end, we develop a deep neural network to identify hands and infer hand joint positions from capacitive frames, and then recover 3D hand poses from the hand-joint positions via a constrained inverse kinematic solver.

Additionally, we design a device setup for capturing high-quality hand-screen interaction data and obtained a more accurate synchronized capacitive video and hand pose dataset. Our method improves the accuracy and stability of 3D hand tracking for capacitive frames while maintaining a compact device setup for remote communication. We validate our scheme design and its superior performance on 3D hand pose tracking and demonstrate the effectiveness of our method in whiteboard-based remote communication.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**.

KEYWORDS

touch, collaboration, input techniques

[*]Also with Microsoft Research Asia.
[†]Work done during the internship at Microsoft Research Asia.
[‡]Corresponding author.

1 INTRODUCTION

Capacitive touchscreens enable users to naturally interact with displayed content via hand fingers, making them widely used as virtual whiteboards in remote communications for efficient user interaction. However, unlike on-site collaborations where participants can simultaneously perceive both the user's gestures and the content displayed on the screen, a local user cannot observe the hand poses of other remote users, which significantly degrades the interaction efficiency and immersive experience of remote communications [5].

A set of methods have been proposed to address this issue. Camera-based solutions [5, 17, 21, 29, 30, 34, 40, 72, 74] employ additional cameras installed on the top of a touchscreen to capture and reconstruct hand poses. While these methods can accurately track hand poses in real time, the cost and size of the extra device setups limit their accessibility and portability, restricting their use in many remote communication scenarios.

Touchscreen-based solutions [1, 15] directly infer hand poses from capacitive images of the touchscreen. On one hand, this approach avoids extra device setup and preserves the accessibility and portability of the device for efficient user interaction. On the other hand, due to the limited range and resolution of the capacitive sensors and the occlusion between different hand parts, the capacitive image only covers a small part of the hands (as shown in Fig. 1), making reconstructing the hand poses from these incomplete capacitive images a challenging problem. Existing methods can only recover static hand poses of single-hand gestures with limited accuracy, hindering their use in real applications where the poses of two hands continuously change during the interaction between hand and screen.

In this paper, we present a method for accurately tracking 3D hand poses from capacitive video frames in real time. Our method estimates continuous 3D poses of two hands interacting with the screen and addresses complex hand gestures as shown in Fig. 1. To achieve this, our method first estimates 3D hand joint positions from input capacitive frames using a deep neural network, then reconstructs hand poses with a constrained inverse kinematic (IK) solver. For the joint position estimator, we adapt an RGB-video-based 3D hand pose estimation network to capacitive image input. The network output includes hand classification results (i.e., left or right) and 3D positions of hand joints. We further incorporate a gated recurrent unit (GRU) architecture into the network, which exploits the temporal information to resolve the ambiguity in hand classification and pose estimation. Finally, the constrained IK solver takes the 3D hand joint positions as input and infers 3D transformations of hand joints for animating a template hand mesh to the resulting 3D hand poses.

For network training, we develop a data acquisition system for capturing capacitive video frames and corresponding 3D hand skeletons from multi-view RGB images and obtain a new dataset. Compared to existing datasets that only consist of static capacitive images and inaccurate hand poses [1], our new dataset is composed of sequences of two-hand gestures with higher pose accuracy and more diverse hand poses.

We evaluate the accuracy of our method using capacitive images of different hand poses and demonstrate its advantages over existing solutions. We also assess the effectiveness of our method in two

remote communication applications. Experimental results indicate that our system enables users to focus more on interaction content, thus improving interaction efficiency. We will release our dataset, code, and model on our project web page.

In summary, our work makes the following contributions:

- We introduce a real-time 3D hand tracking method from capacitive frames that provides temporally consistent and accurate 3D poses of two hands when interacting with a touchscreen.
- We introduce the first capacitive video dataset for 3D hand tracking, which consists of capacitive video clips and corresponding accurate 3D hand poses. The dataset enables our method and facilitates future research on capacitive-based hand pose estimation.
- We apply our method in two whiteboard-based remote communication applications on the touchscreen, which efficiently enhances the user experience while preserving the portability of the device.

2 RELATED WORK

Our work is related to touchscreen technology, hand pose estimation and remote whiteboard interactions. Below we review related work in these areas.

2.1 Touchscreen and Touch Sensing Technology

Touchscreen technology provides an integrated and intuitive solution that seamlessly combines display and user input. The history of touchscreens can be traced back to the pioneering work of E.A. Johnson [33] in 1965. Johnson devised an innovative way to place an array of capacitive sensitive electrodes on the surface of the CRT screen, effectively creating the basis for finger-actuated touchscreens. Over the subsequent decades, a variety of touch sensing technologies were invented for different application scenarios, including capacitive [18, 60], resistive [56], acoustic wave [55], optical [23] and dense optical sensor array [31]. R.A. Boie [8] introduced the first multi-touch screen utilizing capacitive sensing in 1984. Further enhancing the function of capacitive touchscreens, Wayne Westerman [80] developed algorithms to accurately detect multi-touch points on the screen. This technology was subsequently acquired by Apple Inc. and gained widespread adoption with the release of the iPhone in 2007. Capacitive touchscreens offer many advantages over other methods, such as multi-touch support, superior optical quality, and enhanced durability [54]. As a result, they have become the industry standard for mobile phones, tablets, and laptops, fundamentally changing the way we interact with digital devices.

In contemporary consumer devices, capacitive touchscreens use integrated circuits with embedded touch detection algorithms to achieve low-latency multi-touch prediction [3]. To ensure accuracy and efficiency, these algorithms are specifically designed to detect isolated fingertips. Therefore, if two fingers are too close together or the touch area is too large, the touch point may not be detected accurately, thereby limiting the range of freestyle interaction with the touchscreen. With the rapid development of machine learning techniques, data-driven methods have become an important tool to

enhance raw touchscreen data processing. These methods help improve the resolution of touch image sensing [48, 69] and further recover detailed hand poses, including finger recognition [25, 39, 75], touch direction estimation [47, 62, 83], gesture recognition [37, 66], pressure estimation [7], or even some other objects placed on the touchscreen [65], thus offering the potential to extend the capabilities of touchscreen interactions. Compared to these methods that focus on improving touch sensing or enhancing captured signals, our method aims to infer the invisible 3D hand poses from the captured capacitive images.

2.2 Hand Pose Estimation

3D hand pose estimation and tracking from RGB [4, 9, 20, 43, 50] or depth images [14, 19, 52, 61, 84] have been extensively studied in the past decades, with remarkable progress achieved through the introduction of deep neural networks. Recent advances in computer vision have inspired novel methods in this domain, such as adopting the Transformer architecture [77] in model design [32, 35, 79], enhancing models with vision-language pre-trained models [41], and training the models using contrastive learning [44, 67]. In addition to methodological improvements, the emergence of new datasets [10, 22, 51, 53, 92] has further enhanced model performance in hand pose estimation. However, challenges still exist in estimating hand poses with limited information [45, 78, 85]. Recent research has started exploring the use of parametric hand models [63] to tackle issues with incomplete data, such as low resolution or occlusions [9]. While these methods can predict reasonable hand poses for occluded hand parts, the accuracy of visible parts may be compromised due to the reduced motion space.

Different from the RGB or depth images that capture the shape and appearance of full hands, capacitive images only record a low-resolution grayscale image of the partial hand regions that touch the screen, which is similar to the pressure-sensing floor in [11]. As a result, estimating the invisible hand poses from capacitive images is a much more challenging task. Chung et al. [16] proposed a method for estimating a hand model from touch points, which depends on pre-detected touch points and is therefore unsuitable for interaction with free hand gestures. Le et al. [38] utilized a convolutional neural network (CNN) to identify individual fingertips directly from raw capacitive touch images. Recent research has sought to estimate gestures directly from incomplete touchscreen data. Choi et al. [15] constructed a hand pose database and subsequently matched the touchscreen image to one of the reference hand poses, employing nonlinear deformation to align the matched reference hand pose with the touchscreen image. However, due to the complexity of hand gestures and the continuous nature of hand movement, predefined hand categories may not cover all possible hand gestures. Ahuja et al. [1] improved this result by developing a CNN encoder and a multi-layer perceptron (MLP) to estimate joint poses from each touch frame, but they did not take advantage of the temporal consistency of frames, and the direct prediction of Cartesian coordinates of joints difficult to precisely align the touch points on the touch image. Moreover, these approaches do not accommodate scenarios in which both hands interact with the touchscreen simultaneously. Our work introduces a novel approach for accurate hand tracking on capacitive touchscreens, addressing

the limitations by enabling interaction with free hand gestures, using heatmap representation for precise touch point alignment (as evidenced by improved accuracy in recent studies [46, 59, 86]), and support for dual-hand scenarios.

Inverse Kinematics (IK) also plays a crucial role in hand pose estimation by recovering the hand skeleton and hand mesh from joint positions. It involves fitting joints to target positions while adhering to the connectivity and geometric constraints of the template hand skeleton [2, 12, 26, 57]. Once the hand skeleton is obtained, the hand mesh can be bound to the skeleton using automatic rigging techniques [6] and deformed with linear blend skinning (LBS) [42]. In our approach, we employ an IK solver with constraints tailored for touchscreen interactions. For hand mesh generation, we utilize the MANO hand model [63], which provides predefined rigging weights learned from data for a realistic representation of the user's hand. By combining IK, LBS, and MANO hand models, our approach effectively recovers hand bones and hand meshes from the predicted joint positions.

2.3 Remote Whiteboard Interaction

Our work is also related to remote whiteboard interaction, specifically enhancing user experience by integrating visual cues of hand gestures. Previous work has validated and demonstrated that displaying hand movements can efficiently enhance remote whiteboard interactions and facilitate remote collaboration [5]. Remote whiteboard has two different settings: typical whiteboard settings [74] and lightboard settings [49]. In a typical whiteboard setting, each user is given a horizontal desktop display showing shared working items. To convey visual cues of the hand gesture, a vertically downward-facing camera is installed above the desktop display to capture the local user's hand movements and transmit the segmented hand image to the remote user in real time [21, 34, 40, 72]. Some systems also include a separate vertical screen to display the remote user's portrait video [17, 29, 30, 76].

Lightboard settings represent another category of remote collaboration setups. In this configuration, the vertical screen displays working items while also displaying a life-size portrait of the remote user, creating the illusion of collaborative work with users on either side on a clear glass surface. Some works use cameras placed behind transparent screens to capture hand movements [27, 71, 73, 81]; however, this approach requires specialized hardware and takes up a lot of space to install. An alternative approach is to synthesize a 3D hand model using hand tracking sensors mounted around or in front of the screen [82, 89, 91].

Compared to existing solutions, our approach eliminates the need for assistive devices to provide visual cues for gestures during remote collaboration and supports both whiteboard and lightboard configurations by leveraging touch screens for hand tracking. Our approach provides a more versatile and simplified solution with the potential to be implemented on a variety of business mobile devices, thereby enhancing remote collaboration environments.

3 METHOD

Fig. 2 provides an overview of our proposed method, which sequentially processes each touch frame as input and generates estimated
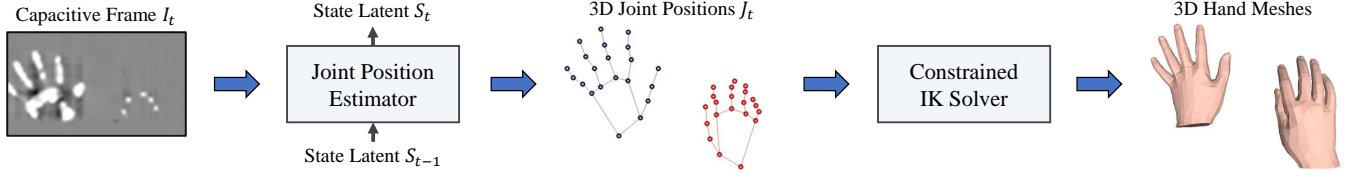
Figure 2: An overview of our method. Given the current capacitive frame I_t and a state latent S_{t-1} aggregated from previous frames, our joint position estimator infers the 3D joint positions J_t of the two hands in the current frame and subsequently updates the state latent to S_t. Following this, a constrained inverse kinematic (IK) solver is employed to reconstruct the 3D hand pose from the 3D hand joint positions and subsequently transform the 3D hand meshes to the current pose.

hand meshes as output. Our approach consists of two key components. The first is a joint position estimator that estimates each joint position for both hands in 3D space. The hand pose history is embedded as a hidden state within the GRU unit of the estimator, enabling temporal consistent prediction of joints not in contact with the touchscreen. The second component is a constrained IK solver that computes parameters for generating hand mesh using the estimated joints. The constrained IK solver also takes into account previous results to minimize noise and ensure smooth hand motion.

3.1 Joint Position Estimator

We use a neural network to estimate the joint position of each hand in the touch image, as illustrated in Fig. 3. For a given touch image of the current frame, the algorithm first determines the existence of each hand within the image and subsequently predicts the 3D coordinates of each joint. To enhance accuracy, we employ a heatmap and depth distribution for each joint, which has been demonstrated to be more accurate than directly predicting the coordinate values [70].

3.1.1 Preprocessing. The touchscreen has a rectangular area of $345mm \times 195mm$ and captures skin contact using a capacitance map. This map is represented as a 71×41 8-bit grayscale image. To process the image, we first normalize it to a range of $[0 - 1]$. When there is no contact, the pixel value is theoretically 0.5, while skin contact increases the value to 1.0. However, factors like fingerprints and smudges can cause variations in these values. Based on our observations, we filter out noise and separate contacting and non-contacting regions by clamping pixel values in the range of $[0.0, 0.6]$ to 0.0. This allows us to identify areas where the skin touches the screen accurately.

As users may touch the screen freely, the projection of their joints might fall outside the image range. To account for this, we expand the image to 128×96 pixels (with a physical range of $620mm \times 456mm$) by adding padding around the original image. We also include an additional channel to distinguish between original and extended pixels. This expansion allows for accurate representation of joint positions using heatmaps regardless of how the hand touches the screen and simplifies the upscaling and downscaling of the image.

3.1.2 Network Structure. Our model is based on U-Net architecture [64, 68] with additional recurrent modules, similar to the one described in [13] (see Fig. 3). The model encodes a two-channel,

128×96 touch image into a $4 \times 3 \times 512$ latent code using five downsampling blocks. The latent code is then refined by a Squeeze-and-Excitation (SE) block [24], which dynamically recalibrates channel-wise features.

The heatmaps, with dimensions of $128 \times 96 \times 42$, are retrieved through five upsampling blocks and a convolution layer. Each upsampling block is followed by a gated recurrent unit (GRU) to record the latent state of previous frames. The heatmaps have 21 channels for each hand, indicating the position of each joint. Further details about the downsampling and upsampling blocks can be found in the Appendix.

The joint projections on the screen are predicted using heatmaps, while the distance of each joint to the screen is predicted separately. The distance is represented by a depth distribution within the range of -1cm to 11cm relative to the screen, which covers the space of human hand interactions with the screen. The range is evenly divided into 48 sections, giving a resolution of $2.5mm$. In our implementation, the final GRU output and latent code are transformed into a 128-element vector using a convolution layer, followed by batch normalization, leaky ReLU activation, and a global average pooling layer. These vectors are concatenated, and the depth distribution is predicted using a fully connected layer with subsequent softmax activation for normalization.

Using the same network architecture as that for depth distribution estimation (only modifying the final activation layer from softmax to sigmoid), we detect the existence of each hand and quantify it as two real numbers within $[0, 1]$.

3.1.3 Loss function. Our loss function

$$L = \lambda_H L_H + \lambda_D L_D + \lambda_B L_B + \lambda_E L_E, \quad (1)$$

consists of four components L_H, L_D, L_B, and L_E.

$$L_H = ||H - H_{gt}||_2, \quad (2)$$

supervises the heatmap prediction of hand joints. Here, H represents the predicted heatmaps, and H_{gt} represents the ground truth heatmaps.

$$L_D = ||d - d_{gt}||_2, \quad (3)$$

supervises the distance of predicted hand joints to the screen. In this case, d refers to the predicted depth distribution of hand joints, and d_{gt} refers to the ground truth.

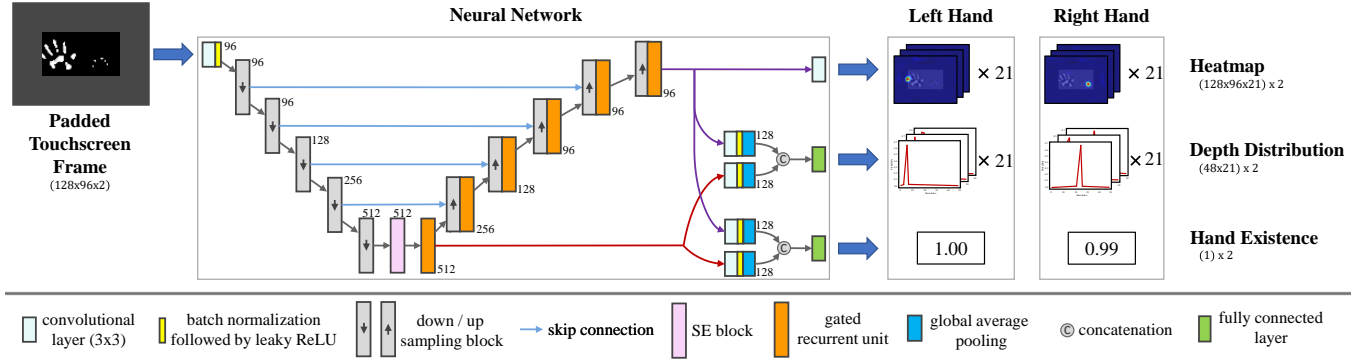$$L_B = ||l - l_{gt}||_2, \quad (4)$$

Figure 3: The network architecture of the joint position estimator.

supervises the bone length of each hand skeleton to achieve a reasonable hand pose. Here, l is the predicted bone length, and l_{gt} is the ground truth value.

$$L_E = BCE(E, E_{gt}), \quad (5)$$

supervises the existence of each hand in the image. BCE is the binary cross-entropy loss function, E is the value indicating the existence of a hand, and E_{gt} is the ground truth value.

In our experiments, we set the weight parameters $\lambda_H = 10$, $\lambda_D = 2$, $\lambda_B = 1$, and $\lambda_E = 0.2$. These weights balance the contribution of each component in the overall loss function.

3.1.4 Training. In our training process, we take advantage of the symmetry between the left and right hands by performing data augmentation. This is done by flipping the image horizontally and swapping the positions of the left and right hands. We use the Adam optimizer [36] to train our model for 1000 epochs, confirming convergence without overfitting by observing the loss curve.

During each epoch, we randomly select N consecutive frames from each touchscreen video as input data. The value of N starts at 2 and increases by 1 after every 20 epochs until it reaches 30. We initialize the learning rate at $1x10^{-3}$ and multiply it by 0.999 every epoch throughout the training process.

3.1.5 Joint Estimation. In this joint estimation algorithm, we perform test-time augmentation to enhance the accuracy of our predictions. This is achieved by inferring both the original touch image and its flipped version. The predicted results are then combined using an averaging technique.

The next step is to obtain the 3D coordinates of each joint from the combined heatmaps and depth distributions. We adopt a classification-regression scheme for coordinates estimation to achieve better accuracy and robustness [58]. For each joint, its 3D coordinate $(\hat{x}, \hat{y}, \hat{z})$ is calculated using the following equation:

$$\begin{cases} \hat{x}, \hat{y} = \arg\max_{x,y} \mathbf{H}_{x,y} \\ \hat{z} = \mathbf{w}^T \mathbf{d}, \end{cases} \quad (6)$$

where $\mathbf{H}$ represents the heatmap of the corresponding joint, $\mathbf{d}$ denotes the 48 basis of depth uniformly sampled in the range of $[-1cm, 11cm]$, and $\mathbf{w}$ is the predicted depth distribution. The existence of each hand is determined by the hand existence coefficient.

If the coefficient is greater than 0.5, it indicates that the corresponding hand is detected in the touch image.

3.2 Constrained Inverse Kinematics Solver

In this section, we aim to predict the hand skeleton and mesh by solving an inverse kinematics problem. The input to this process is the 3D coordinates of each individual joint, which are the output of the hand joint estimator. To maintain a natural hand pose and obtain the hand shape, we apply constrained inverse kinematics to the MANO hand model [63]. This approach differs from parametric hand models [9, 63], as it operates directly on the joints, resulting in better touch accuracy.

The MANO model of each hand consists of two sets of parameters: pose parameters $\Theta \in \mathbb{R}^{48}$, which represent the rotation angles of each joint in root centered coordinate to control the hand skeleton and deform the hand mesh using linear blend skinning, and shape parameters $\beta \in \mathbb{R}^{10}$, which reconstruct the hand shape in the rest state. Based on the anatomical structure of the real hand, we confine the pose parameters of each hand as $\theta \in \mathbb{R}^{26}$, including 6 DoFs for root, 2 DoFs for CMC joint of the thumb, 2 DoFs for MCP joints of other fingers and 1 DoF for all other 10 joints. We denote the joints of the MANO hand model as $\mathbf{f}(\theta, \beta)$. The optimization goal is to match the estimated joint position $\mathbf{J}$ via the following energy function equation:

$$\mathbf{E} = \arg\min_{\theta, \beta} \|\omega \circ (\mathbf{J} - \mathbf{f}(\theta, \beta))\|_2. \quad (7)$$

In this equation, ω represents the weight of each joint. We assign a higher weight to the fingertip joints detected as touching the screen to improve touch point accuracy. In our implementation, $\omega = 2$ for touched fingertip joints and $\omega = 1$ for other joints. To ensure the hand pose is plausible, the following constraints must be met:

$$\theta_{\mathbf{min}} \leq \theta \leq \theta_{\mathbf{max}}, \quad (8)$$

$$\mathbf{f_z}(\theta, \beta) \geq \mathbf{z_0}. \quad (9)$$

Here, Eq. 8 constrains the angle of each joint to the available motion range of the human hand, and Eq. 9 enforces that the fingers
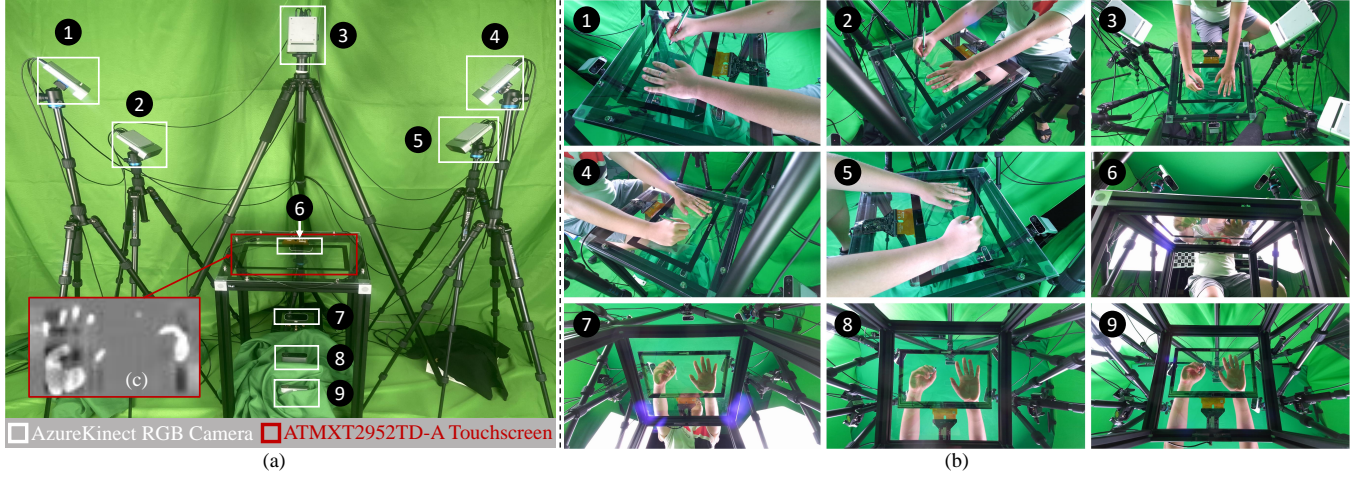
Figure 4: The device setup of our system for capturing ground-truth hand poses and corresponding capacitive images. (a) Our system consists of one capacitive touchscreen and nine RGB cameras, which are synchronized to capture RGB images of hands from different views along with the associated capacitive images. (b) The nine RGB images of two hands on the touchscreen. (c) The capacitive images of the two hands, captured by the touchscreen at the same time instance of the RGB images.

do not penetrate the touchscreen. z_0 represents the joint height when touching the screen.

We employ sequential quadratic programming (SQP) to solve the constrained nonlinear optimization problem. The initial value is carried from the last frame. To maintain a consistent hand shape, we update β for the first 20 frames when all five fingers are detected to have touched the screen. After that, we fix the hand shape by fixing β and perform hand tracking by updating θ only.

In our implementation, we set $z_0 = 5mm$ in Eq. 9, allowing the generated hand mesh to penetrate the screen slightly. This design choice is based on the fact that in reality, skin deforms when fingers touch the screen. We simulate the effect of skin being flattened by the screen through post-processing, projecting the penetrating mesh vertices onto the screen. This simple method effectively mimics the skin flattening effect, providing a stronger sense of realism, which has been confirmed in lightboard applications (Sec. 6.2).

4 DATASET CONSTRUCTION

A dataset that consists of diverse and accurate hand poses and corresponding capacitive images is critical for developing and evaluating learning-based pose-tracking methods. Prior research [1] leveraged LeapMotion for capturing 3D hand skeletons. However, the accuracy of the resulting hand gestures and finger positions is inadequate [89]. The capacitance value of the touch screen exhibits a complex relationship with hand contact and is influenced by external factors such as environmental conditions and palm humidity, making data synthesis challenging. To tackle these challenges, we designed a data acquisition system employing multi-view stereo techniques, allowing us to obtain high-precision 3D hand pose data that is accurately aligned with the touchscreen. Based on this device setup, we have captured a dataset comprising 13.4 hours of varied hand-screen interaction clips, featuring both single and dual-hand engagements, recorded from 16 participants.

4.1 Apparatus

Our data capture system is shown in Fig. 4 (a). The touchscreen is positioned on a table with a transparent surface. To capture the hand from various directions, we employ nine cameras in our setup. We use the AzureKinect RGB camera for its high image quality and the ability to synchronize multiple cameras through cables. All sensors are connected to a single PC via USB.

The cameras are arranged to face the touchscreen from different directions, with five looking downward and four looking upward. The upward-looking cameras under the touchscreen can directly track fingers that may be occluded by the hand during gestures such as writing. The intrinsic and distortion coefficients can be accessed directly from the camera using the sensor SDK. The extrinsic is calibrated using a chessboard and the method proposed in [90]. We then capture an image of the touchscreen from each camera, manually label the four corners of the touchscreen in each image, and calculate their 3D coordinates using triangulation. This process ensures the entire system is calibrated.

To compute the accurate 3D hand pose, all nine cameras recorded images in 1080p (1920×1080) resolution at 30FPS. Since the capacitive frames are captured at 15FPS, we align the RGB frames to the capacitive frames by linearly interpolating the two RGB frames that are closest to each capacitive frame according to their timestamps.

4.2 MVS Hand Pose Estimation

We perform hand pose estimation from the captured multiview images using the method proposed in [28]. First, we detect 2D hand joints in each frame with the hand tracking method in Mediapipe [88], which can track both hands using the 21 joints model with sub-pixel accuracy. Note that although Mediapipe can predict 3D hand poses, the depth of each joint estimated from a single view is not accurate and thus is ignored in our implementation.
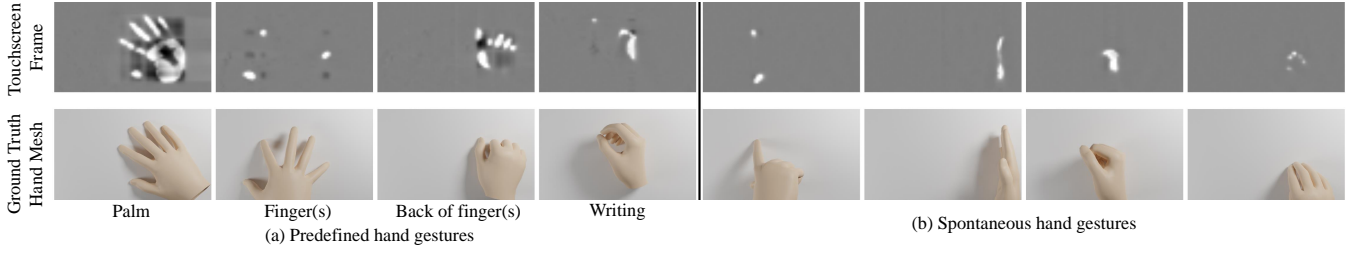
Figure 5: Depiction of the hand gestures in our dataset. During predefined gestures, participants perform hand movements within each category as shown in (a). For free hand movement, participants interchange freely among predefined categories and perform spontaneous hand gestures as shown in (b).

However, detection errors may occur, such as misclassifying left and right hands, misdetecting hands, or having significant joint estimation errors when fingers are occluded in a particular view. For robust 3D hand tracking, we need to filter these failure cases. To this end, we check the correctness of hand h_i detected in one camera i by checking the alignment of each joint of hand h_j in another camera j. We average the distance between rays of the same joint from camera i and j by,

$$D(h_i, h_j) = \frac{1}{21} \sum_{k=1}^{21} \left| (\mathbf{O_i} - \mathbf{O_j}) \cdot \frac{\mathbf{r_i^k} \times \mathbf{r_j^k}}{\left| \mathbf{r_i^k} \times \mathbf{r_j^k} \right|} \right|, \quad (10)$$

Where O_i and O_j are optical centers of camera i and j, $\mathbf{r}_i^k$ and $\mathbf{r}_j^k$ are rays of joint k in camera i and j. In our implementation, if $D(h_i, h_j)$ is smaller than a threshold ($1cm$ in our implementation), then the hands h_i and h_j are considered to be consistent in 3D space. We filter all hands that have less than 3 consistent hands from all other cameras.

Finally, for all remaining hands with high confidence, we compute their joints in 3D space via triangulation. For each joint, its 3D coordinate $\hat{\mathbf{P}}$ is calculated by minimizing the distance to rays $\mathbf{r}_i$ corresponding to the joint point in each camera i, as given by:

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \sum_i \left\| (\mathbf{P} - \mathbf{O}_i) \times \frac{\mathbf{r}_i}{|\mathbf{r}_i|} \right\|_2, \quad (11)$$

where $\mathbf{O}_i$ represents the optical center of camera i. The ray $\mathbf{r}_i$ is calculated by unprojecting the detected 2D joint point on image i using the intrinsic parameters and distortion coefficients of camera i.

4.3 Data Acquisition and Processing

4.3.1 Hand Gestures. Our dataset encompasses a set of capacitive video clips with corresponding 3D hand poses and movements that are frequently used in touch-screen-based user interactions. As shown in Fig. 5(a), we collect four categories of pre-defined hand gestures with different hand contacts with the screen: the palm, the finger(s), the back of the fingers while in a bent position, and holding a stylus for writing. For each gesture type, the participants were guided to naturally hold the hand gesture and move the hand to traverse the entire screen at a regular pace, occasionally extending partially beyond the touch screen area. For finger gestures, the participants pressed one or more fingers on the screen and moved

their hands, as well as switched their fingers on the screen in a manner similar to piano playing. For the writing gesture, the participants followed their conventional writing postures to write or draw across the entire screen. To enhance gesture diversity and ensure a smooth transition between different actions, participants were also asked to perform free hand movements, including transitioning between predefined gestures, as well as spontaneous actions, as shown in Fig.5 (b). While the four categories of pre-defined hand gestures ensure the coverage of most common hand gestures, free hand movement furnishes more adaptable hand movement data and enhances the variety of hand gestures.

During the data acquisition phase, each participant was asked to perform each type of gesture with the right hand, left hand, and two hands. For the two-hand data capturing, we instructed the participants to position their hands on the screen as naturally as possible. The two hands perform the same type of gesture for pre-defined gesture capturing and both move freely in any combination for free gesture capturing. For writing with two hands, the participants performed the writing gesture with their dominant hand while placing the other hand naturally on the screen. Furthermore, we advised the participants to minimize substantial overlaps of the hands during the capturing, without deliberately avoiding minor overlaps.

4.3.2 Participants. Our dataset was contributed to by sixteen participants (9 males and 7 females, aged 19 to 45, mean = 28), covering a broad spectrum of hand sizes (length: 156mm to 195mm, mean = 177mm, width: 70mm to 92mm, mean = 83mm). This diversity ensures our findings are applicable to a wide user base.

4.3.3 Data Acquisition Strategy. We capture hand movement data in sessions. In each session, we start with twelve clips of right-hand gestures, followed by eight clips of the left hand and eight clips of both hands, in a predetermined sequence. For the right hand, we capture 2 clips of each predefined hand gesture and 4 clips of spontaneous hand gestures. For the left and both hands, we capture 1 clip of each predefined hand gesture and the rest are spontaneous hand gestures. Given that all participants were right-handed, their right-hand gestures were more natural and flexible, we captured more right-hand data. Due to the symmetrical nature of hands, left-hand data could be effectively augmented through mirroring right-hand data in post-processing.

For each clip, the capture takes 30 seconds of hand movement data, which contains 900 synchronized video frames from the camera array and approximately 450 frames of touchscreen data. After capturing, the system takes about 30 seconds to dump the clip data to a hard drive and participants can take a rest and get ready for the next hand gesture. For each participant, we capture 4 sessions, which take around 2 hours.

Hand	Right					Left					Both				
	P	F	B	W	Free	P	F	B	W	Free	P	F	B	W	Free
Gesture	117	118	117	115	235	59	59	58	59	231	57	57	53	47	226
Sum			702					466					440		

Table 1: The distribution of our dataset in terms of the number of clips for each gesture, represented by Palm (P), Finger(s) (F), Back of finger(s) (B), Writing (W), and free gestures, for the right, left, and both hands.

4.3.4 Data Processing and Filtering. After data acquisition, we estimate the ground truth 3D hand poses with an offline data processing framework, which takes around 9 minutes for each clip. Quality assurance measures were stringent, with any data compromised by calibration errors, detection inaccuracies, or file corruption promptly excluded, ensuring the dataset's integrity.

Our data acquisition and processing pipeline culminated in a robust dataset of 1608 clips, which is approximately 13.4 hours of a variety of hand movements. The gesture distribution within our dataset is listed in Table. 1. Free hand movement occupies 43% of the whole dataset, ensuring rich behavior of hand motion.

5 METHOD EVALUATION

To evaluate the effectiveness of our hand-tracking scheme, we compared the accuracy of our method with other state-of-the-art approaches, evaluated system performance with both single-hand and two-hand interaction sequences, and conducted an ablation study to verify the contribution of each system component.

5.1 Device and System Performance

We utilize the Microchip ATMXT2952TD-A Evaluation Kit to acquire raw touch images. This kit features a 15.6-inch transparent touchscreen assembly, capable of capturing 8-bit 71×41 raw capacitive touch images at a rate of approximately 15fps and simultaneously detecting up to 16 touch points. The touchscreen is employed in both the data capture system and the whiteboard interaction prototype system.

Our hand-tracking method achieves a performance of 32fps on a PC equipped with an Intel Core i9-10980XE CPU, 64GB memory, and a Nvidia GeForce RTX 4090 GPU. For each input frame, the joint estimation network takes about 13ms to derive the joint position and the inverse kinematics takes about 18ms to infer the 3D hand pose. In our current implementation, the frame rate of the whole system is 15fps and limited by the frame rate of the capacitive imaging.

5.2 Method Validation and Comparison

We evaluate the accuracy and robustness of our method for various hand gestures and participants and compare our method with TouchPose [1], which is the state-of-the-art method for estimating hand poses from capacitive images.

Because TouchPose's dataset in [1] only comprises static images instead of motion sequences, it cannot be directly used for training and testing our method. Additionally, the TouchPose is designed for estimating single-hand gestures only. To facilitate a fair comparison, we construct a right-hand dataset that contains all 576 right-hand sequences in our dataset, and train and test the TouchPose and our model on this right-hand dataset. In particular, the right-hand dataset includes sequences captured from 12 participants [1]. We subdivide the sequences of each participant into 4 sessions, each of which consists of 8 sequences of 4 predefined gesture types (two for each predefined gesture type) and 4 free movement sequences.

Testing Setup. We follow [1] to evaluate the accuracy and generalizability of our method and compare its performance with Touch-Pose with the following three setups:

- **P1** is a 4-fold cross-validation for evaluating the model's performance, in which three sessions of all the participants are used for model training and the remaining one session is used for testing.
- **P2** is a 3-fold cross-validation for evaluating the model's robustness to hand gestures of new users, in which we category 12 participants into three groups and use all sequences of the two participant groups for model training and the data from the remaining group for testing.
- **P3** is a 5-fold cross-validation for evaluating the model's robustness to new gestures, in which we train the model with the data of four out of five hand gesture types and test the model with the data of the remaining one gesture type. The free movements are counted as one gesture type.

Error Metrics. We employed the following error metrics to measure the accuracy of the resulting 3D hand poses inferred by our method and TouchPose:

- **End-point-error (EPE)** computes the average Euclidean distance between the reconstructed hand joint positions and the ground truth ones, measuring the alignment accuracy of the entire hand. We further assess the error parallel to the screen surface (EPE_{xy}) and the error in distance to the screen (EPE_z).
- **End-point-error of visible finger (EPE^v)** measures the end-point-error for joints of fingers that at least one joint is visible on the touchscreen. This metric is also broken down into errors in two directions, denoted by EPE^v_{xy} and EPE^v_z. In remote interaction applications, EPE^v_{xy} can directly reflect the degree of alignment between the hand and the interactive content displayed on the screen.

Results and Analysis. We show the results of our model and retrained TouchPose in Table 2. Compared with the metric values reported in the original TouchPose paper [1], the hand tracking

---

[1] We exclude the data of 4 participants that consists of fewer right-hand sequences than other participants for conducting the three cross-validation evaluations.
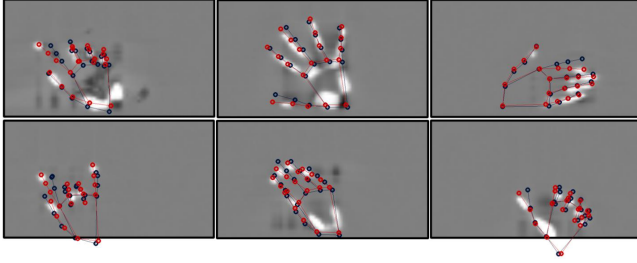
Figure 6: Visual comparison of the projected joint error EPE_{xy} between TouchPose and our method across various hand poses (best viewed on screen). By projecting the joints predicted by our method (red) onto the touch image and comparing them to TouchPose (blue), it is evident that the hand joints predicted by our method exhibit superior alignment with the touch image.

accuracy has been significantly improved by retraining using our larger and more precise dataset of hand movements. This underscores the value of our dataset in improving hand tracking accuracy.

	Method	EPE ↓	EPE_{xy} ↓	EPE_z ↓	EPE^v ↓	EPE^v_{xy} ↓	EPE^v_z ↓
P1	Ours	**8.85 (3.10)**	**6.39 (2.69)**	**4.98 (1.36)**	**7.90 (2.53)**	**5.58 (2.22)**	**4.59 (1.17)**
	TouchPose	11.3 (5.82)	9.19 (4.75)	5.16 (3.00)	10.5 (5.35)	8.56 (4.22)	4.70 (2.99)
P2	Ours	**10.4 (3.47)**	**8.03 (2.96)**	5.21 **(1.64)**	**8.84 (2.82)**	**6.70 (2.54)**	4.63 **(1.25)**
	TouchPose	12.9 (8.01)	11.0 (6.78)	**5.17** (3.62)	11.5 (7.39)	9.90 (6.11)	**4.51** (3.61)
P3	Ours	**11.8 (5.56)**	**9.03 (4.68)**	**5.99 (2.52)**	**9.67 (3.95)**	**7.46 (3.45)**	**4.84 (1.65)**
	TouchPose	15.0 (8.38)	12.5 (7.19)	6.57 (3.67)	13.1 (7.19)	10.9 (6.09)	5.59 (3.35)

Table 2: Comparative analysis of hand-tracking accuracy and standard deviation (in brackets) between our method and retrained TouchPose across different evaluation protocols (P1, P2, P3). The unit of all values in the table are millimeters (mm).

By incorporating temporal information and employing heatmaps for joint position prediction, our method further improved hand-tracking accuracy and reduced deviation, particularly in the xy direction, which is parallel to the screen. This indicates the efficacy of precise joint prediction using heatmaps. A visual comparison of EPE_{xy} between our method and TouchPose for various gestures can be found in Fig. 6. For the z-direction error, our method's performance is on par with (or slightly better than) TouchPose, highlighting the inherent challenge in predicting the accurate pose of non-visible hand parts, especially the invisible fingers.

Temporal information not only facilitates hand tracking but also aids in resolving ambiguities in hand gesture tracking. For example, when a single fingertip contacts the screen, utilizing temporal data helps accurately determine the finger's identity by tracking the alternation of different fingertips. We show this tracking improvement in the supplemental video (2:40-3:05). This kind of hand gesture ambiguity cannot be resolved by the conventional temporal filter. Our evaluation across the three protocols and comparison with TouchPose demonstrates the accuracy and generalizability of our method in hand gesture recognition.

Hand	Gesture	EPE	EPE_{xy}	EPE_z	EPE^v	EPE^v_{xy}	EPE^v_z	HEA(%)
Right	P	9.22	7.72	3.85	8.35	7.03	3.47	99.9
	F	12.2	9.69	5.99	10.6	8.56	5.08	99.9
	B	12.4	10.1	5.61	11.3	9.28	5.09	99.8
	W	16.9	14.2	7.03	14.8	12.3	6.43	99.9
	Free	11.7	9.42	5.47	10.5	8.58	4.87	99.8
Left	P	8.52	6.96	3.82	7.34	5.98	3.34	97.9
	F	10.6	8.35	5.32	9.29	7.21	4.77	99.8
	B	12.4	9.88	5.97	11.3	9.00	5.45	99.1
	W	14.1	10.8	7.49	9.77	7.59	5.01	99.4
	Free	11.4	9.03	5.60	10.3	8.19	5.08	99.3
Both	P	10.8	8.99	4.57	9.59	7.99	4.08	100
	F	12.1	9.52	6.02	10.1	8.17	4.91	100
	B	12.5	10.1	5.66	11.9	9.76	5.31	100
	W	14.0	11.4	6.27	11.7	9.58	5.16	100
	Free	11.5	9.33	5.40	10.6	8.65	4.85	100
Avg		11.8	9.46	5.58	10.4	8.35	4.86	99.7

Table 3: Evaluation of hand-tracking performance across different hands and gestures (P: Palm, F: Finger(s), B: Back of finger(s), W: Writing). The unit of all values for EPE are millimeters (mm).

5.3 More Evaluations

Our system supports interaction with the touchscreen using either one or both hands. We evaluated our full model's performance for inferring the gestures of the right, left, and both hands. An additional metric, Hand-existence-accuracy (HEA), the correct rate of hand existence, was introduced to measure the accuracy with which our method can identify the presence of either hand.

To simulate the performance of the method in real scenarios where the user is always not the one in the training dataset, we use the data of randomly-selected 12 participants for training and the sequences of the remaining 4 participants for testing. This is similar to the P2 testing setup described above without n-fold cross-validation. The accuracy of our model for each gesture type and hand-type (left-hand, right-hand, and two-hand) are listed in Table 3.

The system achieves an overall accuracy of 11.8mm. For fingers in contact with the touchscreen surface, the accuracy reached 10.4mm, with EPE^v_{xy}=8.35mm (error of contacted fingers in the direction parallel to the screen surface). As a result, the shifting of the rendered fingers is hardly noticeable in our remote whiteboard applications.

The accuracy of hand tracking varies with different gestures and the contact area with the screen. Generally, the larger the contact area that can display the overall posture of the hand, the smaller the error. For example, when the palm touches the screen (i.e. the gesture P), the fingers are spread out on the screen, and the capacitive image can reflect the overall shape of the palm, resulting in small errors in both horizontal and vertical directions. When only part of the fingertips or the back of the fingers contacts the screen (i.e. gesture F and B), the posture of the palm and the lift up fingers need to be estimated, leading to a decrease in hand tracking accuracy. For writing, gestures among individuals exhibit

(a) The device on
the speaker's side.

(b) The whiteboard on
the remote audience's side.

(c) The lightboard on
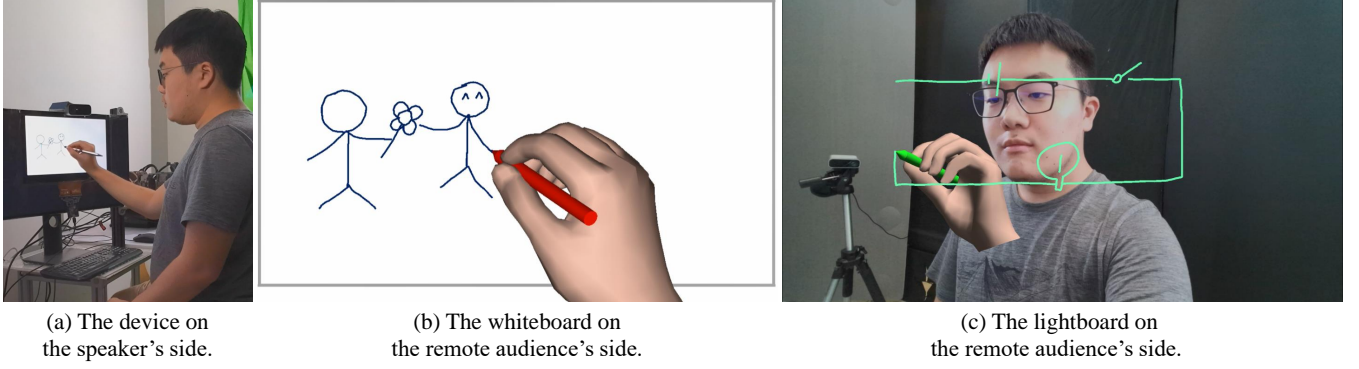the remote audience's side.

Figure 7: The two remote communication applications for method evaluation. (a) The device setup on the speaker's side. (b) A screenshot of the whiteboard application, where both remote audiences and the speaker are positioned on the same side of a virtual whiteboard. (c) A screenshot of the lightboard application where the speaker and remote audiences are on opposite sides of a semi-transparent virtual whiteboard, enabling audiences to observe the speaker's expressions and interactions with the whiteboard.

Method	EPE(mm) ↓	HEA(%) ↑
W/o Temporal info (GRU)	13.9	96.3
W/o SE block	12.9	99.6
W/o test-time augmentation	13.0	99.5
Our full model	**11.8**	**99.7**

Table 4: The results of the ablation study. We evaluate the contribution of each module to the accuracy of the estimated hand pose.

variability, while the capacitive image only captures the palm's edge. Consequently, our method is difficult to accurately reconstruct the specific gesture of holding a pen based solely on the capacitive image.

In terms of end-point-error metric, a consistent trend for accuracy across various hand gestures is observed for both single-hand and dual-hand scenarios, with the left hand's accuracy marginally exceeding that of the right hand. This discrepancy can be attributed to the fact that all participants in our dataset are right-handed, which results in a reduced range of motion for the left hand due to its lesser flexibility.

In terms of the accuracy of determining hand presence, the judgment for both hands is very stable when both are in contact with the screen. However, when only one hand touches the screen, it becomes necessary to identify which hand it is, and the success rate of recognizing the right hand is slightly higher than that of the left hand. This may be because the dataset contains more data for the right hand than for the left hand, thus allowing for more accurate recognition of right-hand movements. These results also indicate that in future work, our dataset should include data from participants who are left-handed, to achieve a balance in the quantity of data and the range of motion for gestures between the left and right hands.

5.4 Ablation Study

We conduct an ablation study by removing each component in our method and retrain the model with the setup described in Section 5.3. We then follow the scheme in Section 5.3 to evaluate each retrained model on testing sequences of 4 participants. As shown in Table 4, each component makes an important contribution to the accuracy of the estimated hand poses. Combining all components, our full model achieves the highest overall performance, highlighting the synergistic effect of the integrated system design.

6 APPLICATIONS

In this section, we show potential applications of our method in remote whiteboard interactions and the user experience of our system. Our prototype whiteboard system setup, as illustrated in Fig. 7 (a), follows a typical tablet configuration, complete with a touchscreen and a frontal camera. We attach the touchscreen to the monitor screen and adjust the display viewport to align with the touchscreen boundaries. An RGB camera is positioned at the border of the touchscreen and serves as the frontal camera.

6.1 Whiteboard

We developed a whiteboard prototype system on the hardware device depicted in Fig.7 (a). On the speaker's side, the experience is similar to that of a traditional whiteboard system, allowing them to write directly on the screen using a stylus. On the audience's side, not only is the whiteboard content displayed, but also the speaker's hand and pen are visible, as illustrated in Fig.7 (b).

Utilizing the touchscreen's ability to simultaneously capture touchpoint coordinates and the capacitive touch image, we calculate the stylus tip's position based on touchpoint coordinates and estimate the hand posture holding the pen using the contact image. Subsequently, we fit a 3D pen model to the hand posture, aligning the pen tip with the touchpoint. This method enables the audience to observe the speaker's writing process on the whiteboard, achieving a visual presenter-like effect.
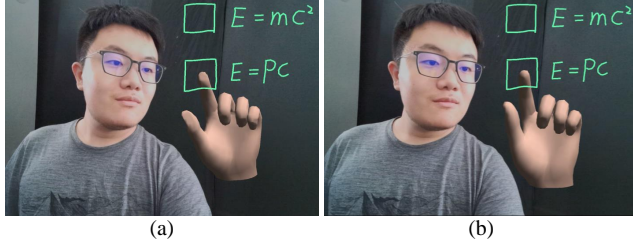
(a)   (b)

Figure 8: Passive hand deformation caused by touch pressure. (a) The rendering of the hand featuring deformed fingertips, simulated by our constrained IK solver. (b) The rendering of the hand without any fingertip deformation.

Our system can estimate the gestures of both hands simultaneously, allowing the speaker to perform operations directly through gestures, such as erasing using a fist. For additional whiteboard interaction examples, please refer to our accompanying video. Our whiteboard system not only provides convenience for speakers but also enhances the visual experience for the audience, making it a valuable asset for presentations and discussions.

6.2   Lightboard

The lightboard application emulates the experience of a speaker and remote audiences situated on opposite sides of a semi-transparent whiteboard, enabling audiences to view the speaker's expressions and interactions with the written content. Conventional lightboards require specialized equipment, but our hand-tracking method on touchscreens enables us to easily create a lightboard experience on touchscreen devices.

In our system, the lightboard employs the same device as the whiteboard mentioned earlier, but utilizes the RGB camera to capture live video of the speaker. Strokes, hands, and the pen are rendered from the backside of the screen, with the video displayed as the background. In this way, the speaker appears to be writing on a transparent glass surface, as illustrated in Fig.7 (c). Notice that when the speaker draws with the stylus on the touchscreen, his eyes are also concentrated on the drawing point. The consistency of hand movements and gaze enhances the speaker's sense of realism and increases the audience's engagement.

To further enhance the visual experience of hand interaction on the lightboard, we employ passive deformation of the hand, enabled by the constrained IK solver. By deforming regions of the hand mesh that touch the screen, the flattened area delivers visual cues that the hand is touching the screen, such as the fingertips, as shown in Fig.8 (a), and the little finger, as depicted in Fig.7 (c).

Given that the content on the glass appears mirrored when observed from the back, we implement a horizontal flip on the synthesized video before displaying it to audiences. This approach is a common solution employed by lightboard devices. Consequently, the displayed content appears normal, while the speaker is flipped and appears to be writing with his left hand (assuming the speaker is right-handed). According to the user experience tests, this mirroring does not negatively impact the overall user experience, maintaining a seamless and professional appearance.
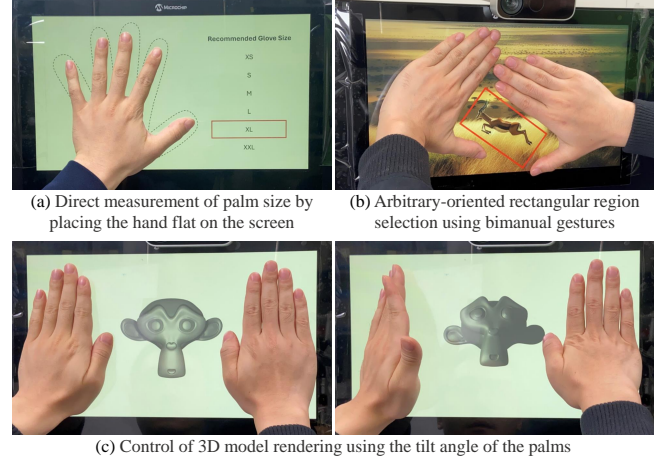

(a) Direct measurement of palm size by placing the hand flat on the screen

(b) Arbitrary-oriented rectangular region selection using bimanual gestures

(c) Control of 3D model rendering using the tilt angle of the palms

Figure 9: Illustrative examples of multimodal gestural interactions using our hand pose estimation method.

6.3   Other Applications

Aside from whiteboard applications, our method's capability for real-time estimation of dual-hand postures on touchscreens unveils several more potential applications. In Fig. 9 (a), we showcase a potential application in e-commerce: simply by placing the palm flat against the touchscreen, one can directly estimate the size of the palm, facilitating the selection of appropriately sized gloves. Additionally, our method enables exceptionally convenient user-computer interactions through gestures. Illustrated in Fig. 9 (b) is the capability to directly delineate a rectangular area of any orientation on the screen using the thumb and index finger of both hands; and in Fig. 9 (c), the capability to manipulate 3D model rendering by altering the palm-to-touchscreen contact angle. The left and right hands independently modulate the model's lighting and rotation, respectively, simplifying the process of examining 3D models for users. We are confident that touchscreens with the ability for real-time gesture estimation will unlock further possibilities in human-computer interaction.

6.4   User Experience

To validate the effectiveness of our approach in enhancing whiteboard and lightboard applications, we conducted an informal user experience test with 16 participants. Before the test, we prepared four video clips showcasing four scenarios: whiteboard and lightboard, with and without hand display. The participants, acting as the audience, compared the hand display modes in each scenario to determine which approach better facilitated their focus on the content written by the presenter. In the whiteboard mode, 13 participants (81%) found that incorporating the hand display improved their attention, while in the lightboard mode, 15 participants (94%) agreed.

In the lightboard mode, we asked participants if they noticed that the displayed video was mirrored, with the speaker appearing to write with their left hand. Only two participants (12%) observed this independently, and they all agreed that the mirroring had no impact on their experience. Comparing the conditions shown with

and without passive hand deformation, all participants agreed that passive hand deformation led them to believe that the hand had touched the screen.

For our current hand display implementation, we utilized a non-photorealistic rendering technique. We also presented an alternative approach where hand images were captured directly through a camera and transmitted in real time to the interactive whiteboard. Among participants, eight (50%) preferred the non-photorealistic rendering for reasons such as privacy protection, consistency with the whiteboard drawing style, and simplicity of texture, which helped them concentrate on the content on the whiteboard. All participants agreed that if a photorealistic hand drawing was not available, our current display method serves as a satisfactory alternative for fulfilling interactive requirements.

In conclusion, the user experience test results demonstrate that our hand-tracking technique significantly enhances participants' attention and engagement in both whiteboard and lightboard modes.

7 LIMITATIONS AND FUTURE WORK

Our designed gestures cover common gestures associated with touchscreen interactions and allow participants to move freely as a supplement, yet it still cannot encompass all possible gestures. Therefore, it fails to accurately track extreme gestures. Currently, our data collection system requires post-processing to calculate the 3D skeleton of the hand. Our hand motion dataset, which captured 13.4 hours of hand motion data, required 32 hours for filming and 250 hours for post-processing. This process does not allow for the timely identification of issues during filming, leading to wasted data capture efforts and low data collection efficiency. To address this issue, we plan to use more gesture categories (similar to [15]) to guide users in capturing hand motion data, and use a real-time hand tracking system based on vision [87] or motion capture systems [38] for real-time quality inspection. This would significantly improve data collection efficiency, making it possible to construct large-scale datasets that include more participants and a greater diversity of gestures.

Our current inverse kinematics solver does not adequately consider scenarios where the hand is holding an object, such as a pen. This approach, which merely positions the pen directly at the hand's location, often results in an unnatural pen-holding posture. Furthermore, we constrain the pen tip's position to the touch points on the screen, leading to noticeable jittering when the pen tip is lifted. For future enhancements, we propose integrating the pen into the kinematic model to more accurately represent gestures involving objects. This modification is expected to produce more natural hand movements, particularly when interacting with items like a pen.

From an application perspective, our current system only displays cartoon-style hands, without showing arms. Investigating the challenge of rendering realistic hand and arm displays, and achieving consistency with the user's body in a lightboard scene, presents a compelling research direction. Our existing whiteboard system operates in a broadcast mode, preventing other users from participating in the interaction. Exploring the design of touchscreen-based whiteboard and lightboard systems that support multi-user participation is a valuable endeavor. Additionally, at present, we simulate

the tablet experience by affixing a touchscreen development kit to the screen. We plan to customize the device driver to enable our system to operate on actual tablets and mobile phones.

8 CONCLUSION

We present a real-time 3D hand-tracking method for estimating hand poses from capacitive touch images. Our system utilizes a Recurrent U-Net network structure to process capacitive images and deliver temporally consistent and accurate hand poses while two hands interact with the screen. We also captured a ground-truth hand movement dataset via a multi-view acquisition system, which is used for network training and improves the pose estimation performance. Our method greatly improves the generality and accuracy of capacitive image-based pose estimation and enhances the user experience of whiteboard-based remote communication systems. We believe that our method could help facilitate more intuitive and seamless human interaction with touchscreens.

ACKNOWLEDGMENTS

REFERENCES

[1] Karan Ahuja, Paul Streli, and Christian Holz. 2021. TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images. In *UIST 2021*. ACM, 997–1009.

[2] Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graph. Model.* 73, 5 (2011), 243–260.

[3] Atmel 2015. *maXTouch 2911-node Touchscreen Controller Datasheet.* Atmel.

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2019. Pushing the Envelope for RGB-Based Dense 3D Hand Pose Estimation via Neural Rendering. In *CVPR 2019*. IEEE, 1067–1076.

[5] Huidong Bai, Prasanth Sasikumar, Jing Yang, and Mark Billinghurst. 2020. A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing. In *CHI 2020*. ACM, 1–13.

[6] Ilya Baran and Jovan Popović. 2007. Automatic rigging and animation of 3d characters. *TOG* 26, 3 (2007), 72–es.

[7] Tobias Boceck, Sascha Sprott, Huy Viet Le, and Sven Mayer. 2019. Force Touch Detection on Capacitive Sensors using Deep Neural Networks. In *MobileHCI 2019*. ACM, 42:1–42:6.

[8] Robert A Boie. 1984. Capacitive impedance readout tactile image sensor. In *Proc. ICRA 1984*, Vol. 1. IEEE, 370–378.

[9] Adnane Boukhayma, Rodrigo Andrade de Bem, and Philip H. S. Torr. 2019. 3D Hand Shape and Pose From Images in the Wild. In *CVPR 2019*. IEEE, 10843–10852.

[10] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *ECCV 2020*, Vol. 12358. Springer, 361–378.

[11] Alan Bränzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Lühne, René Meusel, Stephan Richter, and Patrick Baudisch. 2013. GravitySpace: tracking users and their poses in a smart room using a pressure-sensing floor. In *CHI 2013*. 725–734.

[12] Samuel R. Buss and Jin-Su Kim. 2005. Selectively Damped Least Squares for Inverse Kinematics. *J. Graph. Tools* 10, 3 (2005), 37–49.

[13] Meixu Chen, Todd Goodall, Anjul Patney, and Alan C. Bovik. 2022. Learning to compress videos without computing motion. *Signal Process. Image Commun.* 103 (2022), 116633.

[14] Jian Cheng, Yanguang Wan, Dexin Zuo, Cuixia Ma, Jian Gu, Ping Tan, Hongan Wang, Xiaoming Deng, and Yinda Zhang. 2022. Efficient Virtual View Selection for 3D Hand Pose Estimation. In *AAAI 2022*. AAAI Press, 419–426.

[15] Frederick Choi, Sven Mayer, and Chris Harrison. 2021. 3D Hand Pose Estimation on Conventional Capacitive Touchscreens. In *MobileHCI 2021*. ACM, 3:1–3:13.

[16] Se-Joon Chung, Junggon Kim, Shangchen Han, and Nancy S. Pollard. 2015. Quadratic Encoding for Hand Pose Reconstruction from Multi-Touch Input. In *Eurographics 2015*. Eurographics Association, 13–16.

[17] Francois Coldefy and Stphane Louis-dit Picard. 2007. DigiTable: an interactive multiuser table for collocated and remote collaboration enabling remote gesture visualization. In *CVPR 2007*. 1–8.

[18] Paul Dietz and Darren Leigh. 2001. DiamondTouch: a multi-user touch technology. In *UIST 2001*. 219–226.

[19] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. 2020. JGR-P2O: Joint Graph Reasoning Based Pixel-to-Offset Prediction Network for 3D Hand Pose Estimation from a Single Depth Image. In *ECCV 2020*, Vol. 12351. Springer, 120–137.

[20] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D Hand Shape and Pose Estimation From a Single RGB Image. In *CVPR 2019*. IEEE, 10833–10842.

[21] Aaron M. Genest, Carl Gutwin, Anthony Tang, Michael Kalyn, and Zenja Ivkovic. 2013. KinectArms: A Toolkit for Capturing and Displaying Arm Embodiments in Distributed Tabletop Groupware. In *CSCW 2013*. ACM, 157–166.

[22] Francisco Gomez-Donoso, Sergio Orts-Escolano, and Miguel Cazorla. 2019. Large-scale multiview 3D hand pose dataset. *Image Vis. Comput.* 81 (2019), 25–33.

[23] Jefferson Y Han. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST 2005*. 115–118.

[24] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *CVPR 2018*. IEEE, 7132–7141.

[25] Zeyuan Huang, Cangjun Gao, Haiyan Wang, Xiaoming Deng, Yu-Kun Lai, Cuixia Ma, Sheng-feng Qin, Yong-Jin Liu, and Hongan Wang. 2024. SpeciFingers: Finger Identification and Error Correction on Capacitive Touchscreens. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 1 (2024), 1–28.

[26] Charles W. Wampler II. 1986. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. *IEEE Trans. Syst. Man Cybern.* 16, 1 (1986), 93–101.

[27] Hiroshi Ishii and Minoru Kobayashi. 1992. ClearBoard: A Seamless Medium for Shared Drawing and Conversation with Eye Contact. In *CHI 1992*. ACM, 525–532.

[28] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. 2019. Learnable Triangulation of Human Pose. In *ICCV 2019*. 7717–7726.

[29] Daisuke Iwai, Ryo Matsukage, Sota Aoyama, Tsuyoshi Kikukawa, and Kosuke Sato. 2018. Geometrically Consistent Projection-Based Tabletop Sharing for Remote Collaboration. *IEEE Access* 6 (2018), 6293–6302.

[30] Shahram Izadi, Ankur Agarwal, Antonio Criminisi, John Winn, Andrew Blake, and Andrew Fitzgibbon. 2007. C-Slate: A Multi-Touch and Object Recognition System for Remote Collaboration using Horizontal Surfaces. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. 3–10.

[31] Shahram Izadi, Steve Hodges, Alex Butler, Darren West, Alban Rrustemi, Mike Molloy, and William Buxton. 2009. Thinsight: a thin form-factor interactive surface technology. *Comm. of the ACM* 52, 12 (2009), 90–98.

[32] Changlong Jiang, Yang Xiao, Cunlin Wu, Mingyang Zhang, Jinghong Zheng, Zhiguo Cao, and Joey Tianyi Zhou. 2023. A2J-Transformer: Anchor-to-Joint Transformer Network for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *CVPR 2023*. 8846–8855.

[33] Eric Arthur Johnson. 1965. Touch display? a novel input/output device for computers. *Electronics Letters* 8, 1 (1965), 219–220.

[34] Sasa Junuzovic, Kori Inkpen, Tom Blank, and Anoop Gupta. 2012. IllumiShare: Sharing Any Surface. In *CHI 2012*. ACM, 1919–1928.

[35] Jakub Kanis, Ivan Gruber, Zdenek Krnoul, Matyás Boháček, Jakub Straka, and Marek Hrúz. 2023. MuTr: Multi-Stage Transformer for Hand Pose Estimation from Full-Scene Depth Image. *Sensors* 23, 12 (2023), 5509.

[36] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

[37] 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 *CHI 2018*. ACM.

[38] Huy Viet Le, Sven Mayer, and Niels Henze. 2018. InfiniTouch: Finger-aware interaction on fully touch sensitive smartphones. In *UIST 2018*. 779–792.

[39] Huy Viet Le, Sven Mayer, and Niels Henze. 2019. Investigating the feasibility of finger identification on capacitive touchscreens using deep learning. In *IUI 2019*. 637–649.

[40] Khanh-Duy Le, Kening Zhu, and Morten Fjeld. 2017. Mirrortablet: Exploring a Low-Cost Mobile System for Capturing Unmediated Hand Gestures in Remote Collaboration. In *MUM 2017*. ACM, 79–89.

[41] Seongyeong Lee, Hansoo Park, Dong Uk Kim, Jihyeon Kim, Muhammadjon Boboev, and Seungryul Baek. 2023. Image-free Domain Generalization via CLIP for 3D Hand Pose Estimation. In *WACV 2023*. IEEE, 2933–2943.

[42] John P Lewis, Matt Cordner, and Nickson Fong. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*. 165–172.

[43] Fanqing Lin, Connor Wilhelm, and Tony R. Martinez. 2021. Two-hand Global 3D Pose Estimation using Monocular RGB. In *WACV 2021*. IEEE, 2372–2380.

[44] Qiuxia Lin, Linlin Yang, and Angela Yao. 2023. Cross-Domain 3D Hand Pose Estimation With Dual Modalities. In *CVPR 2023*. 17184–17193.

[45] Xiangbo Lin, Yibo Li, Yidan Zhou, Yi Sun, and Xiaohong Ma. 2022. FSR: a feature self-regulation network for partially occluded hand pose estimation. *Signal Image Video Process.* 16, 5 (2022), 1187–1195.

[46] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. 2021. Rethinking the heatmap regression for bottom-up human pose estimation. In *CVPR 2021*. 13264–13273.

[47] Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *ISS 2017*. ACM, 220–229.

[48] Sven Mayer, Xiangyu Xu, and Chris Harrison. 2021. Super-resolution capacitive touchscreens. In *CHI 2021*. 1–10.

[49] Sarah McCorkle and Paul Whitener. 2020. The Lightboard: Expectations and Experiences. *International Journal of Designs for Learning* 11, 1 (2020), 75–83.

[50] Hao Meng, Sheng Jin, Wentao Liu, Chen Qian, Mengxiang Lin, Wanli Ouyang, and Ping Luo. 2022. 3D Interacting Hand Pose Estimation by Hand De-occlusion and Removal. In *ECCV 2022*, Vol. 13666. Springer, 380–397.

[51] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *ECCV 2020*, Vol. 12365. Springer, 548–564.

[52] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. 2019. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *TOG* 38, 4 (2019), 49:1–49:13.

[53] Battushig Myanganbayar, Cristina Mata, Gil Dekel, Boris Katz, Guy Ben-Yosef, and Andrei Barbu. 2018. Partially Occluded Hands: - A Challenging New Dataset for Single-Image Hand Pose Estimation. In *ACCV 2018*, Vol. 11365. Springer, 85–98.

[54] Hyoungsik Nam, Ki-Hyuk Seol, Junhee Lee, Hyeonseong Cho, and Sang Won Jung. 2021. Review of Capacitive Touchscreen Technologies: Overview, Research Trends, and Machine Learning Approaches. *Sensors* 21, 14 (2021), 4776.

[55] Takaaki Nara, Masaya Takasaki, Taro Maeda, Toshiro Higuchi, Shigeru Ando, and Susumu Tachi. 2001. Surface acoustic wave tactile display. *IEEE computer graphics and applications* 21, 6 (2001), 56–63.

[56] George S HurstJames E Parks. 1970. Electrical sensor of plane coordinates. US Patent US3662105A.

[57] Alexandre N. Pechev. 2008. Inverse Kinematics without matrix inversion. In *ICRA 2008*. IEEE, 2005–2012.

[58] Silvia L Pintea, Yancong Lin, Jouke Dijkstra, and Jan C van Gemert. 2023. A step towards understanding why classification helps regression. In *ICCV*. 19972–19981.

[59] Haoxuan Qu, Li Xu, Yujun Cai, Lin Geng Foo, and Jun Liu. 2022. Heatmap distribution matching for human pose estimation. *NIPS* 35 (2022), 24327–24339.

[60] Jun Rekimoto. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI 2002*. 113–120.

[61] Pengfei Ren, Haifeng Sun, Jiachang Hao, Jingyu Wang, Qi Qi, and Jianxin Liao. 2022. Mining Multi-View Information: A Strong Self-Supervised Framework for Depth-based 3D Hand Pose and Mesh Estimation. In *CVPR 2022*. IEEE, 20523–20533.

[62] Simon Rogers, John Williamson, Craig D. Stewart, and Roderick Murray-Smith. 2011. AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation. In *CHI 2011*. ACM, 2575–2584.

[63] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graph.* 36, 6 (2017), 245:1–245:17.

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015*. Springer, 234–241.

[65] Martin Schmitz, Florian Müller, Max Mühlhäuser, Jan Riemann, and Huy Viet Le. 2021. Itsy-bits: Fabrication and recognition of 3d-printed tangibles with small footprints on capacitive touchscreens. In *CHI 2021*. 1–12.

[66] Robin Schweigert, Jan Leusmann, Simon Hagenmayer, Maximilian Weiß, Huy Viet Le, Sven Mayer, and Andreas Bulling. 2019. KnuckleTouch: Enabling Knuckle Gestures on Capacitive Touchscreens using Deep Learning. In *Proceedings of Mensch und Computer 2019*. ACM, 387–397.

[67] Adrian Spurr, Aneesh Dahiya, Xi Wang, Xucong Zhang, and Otmar Hilliges. 2021. PeCLR: Self-Supervised 3D Hand Pose Estimation from monocular RGB via Equivariant Contrastive Learning. In *ICCV 2021*. 11230–11239.

[68] Benedict Steuerlein and Sven Mayer. 2022. Conductive fiducial tangibles for everyone: A data simulation-based toolkit using deep learning. *Proceedings of the ACM on Human-Computer Interaction* 6, MHCI (2022), 1–22.

[69] Paul Streli and Christian Holz. 2021. Capcontact: Super-resolution contact areas from capacitive touchscreens. In *CHI 2021*. 1–14.

[70] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. 2018. Integral Human Pose Regression. In *ECCV 2018 (Lecture Notes in Computer Science, Vol. 11210)*. Springer, 536–553.

[71] Kar-Han Tan, Ian Robinson, Ramin Samadani, Bowon Lee, Dan Gelb, Alex Vorbau, Bruce Culbertson, and John Apostolopoulos. 2009. ConnectBoard: A remote collaboration system that supports gaze-aware interaction and sharing. In *2009*

IEEE International Workshop on Multimedia Signal Processing. 1–6.

[72] Anthony Tang, Carman Neustaedter, and Saul Greenberg. 2007. VideoArms: Embodiments for Mixed Presence Groupware. In *People and Computers XX — Engage.* Springer London, 85–102.

[73] John C. Tang and Scott Minneman. 1991. VideoWhiteboard: Video Shadows to Support Remote Collaboration. In *CHI 1991.* ACM, 315–322.

[74] Hervé Thily. 2010. Using a digital presenter as a mixed solution to teaching and training. *Teaching English with Technology* 10, 2 (2010), 69–78.

[75] Ying-Chao Tung, Mayank Goel, Isaac Zinda, and Jacob O Wobbrock. 2018. RainCheck: overcoming capacitive interference caused by rainwater on smartphones. In *ICMI 2018.* 464–471.

[76] Baris Unver, Sarah A McRoberts, Sabirat Rubya, Haiwei Ma, Zuoyi Zhang, and Svetlana Yarosh. 2016. ShareTable application for HP sprout. In *CHI EA 2016.* 3784–3787.

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2019. Attention is All you Need. In *NIPS 2017.* 5998–6008.

[78] Rong Wang, Wei Mao, and Hongdong Li. 2023. Interacting Hand-Object Pose Estimation via Dense Mutual Attention. In *WACV 2023.* IEEE, 5724–5734.

[79] Yilin Wen, Hao Pan, Lei Yang, Jia Pan, Taku Komura, and Wenping Wang. 2023. Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos. In *CVPR 2023.* 21243–21253.

[80] Wayne Westerman. 1999. *Hand tracking, finger identification, and chordic manipulation on a multi-touch surface.* Citeseer.

[81] Andrew D. Wilson. 2004. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. In *In ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces.* ACM Press, 69–76.

[82] Erroll Wood, Jonathan Taylor, John Fogarty, Andrew Fitzgibbon, and Jamie Shotton. 2016. ShadowHands: High-Fidelity Remote Hand Gesture Visualization Using a Hand Tracker. In *ISS 2016.* ACM, 77–84.

[83] Robert Xiao, Julia Schwarz, and Chris Harrison. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. In *ITS 2015.* ACM, 47–50.

[84] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. 2019. A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation From a Single Depth Image. In *ICCV 2019.* IEEE, 793–802.

[85] Hao Xu, Tianyu Wang, Xiao Tang, and Chi-Wing Fu. 2023. H2ONet: Hand-Occlusion-and-Orientation-Aware Network for Real-Time 3D Hand Mesh Reconstruction. In *CVPR 2023.* 17048–17058.

[86] John Yang, Hyung Jin Chang, Seungeui Lee, and Nojun Kwak. 2020. SeqHAND: RGB-sequence-based 3D hand pose and shape estimation. In *ECCV 2020.* Springer, 122–139.

[87] Zhipeng Yu and Yangang Wang. 2022. A Synchronized Multi-view System for Real-Time 3D Hand Pose Estimation. In *CAAI International Conference on Artificial Intelligence.* Springer, 588–593.

[88] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv preprint arXiv:2006.10214* (2020).

[89] Yizhong Zhang, Zhiqi Li, Sicheng Xu, Chong Li, Jiaolong Yang, Xin Tong, and Baining Guo. 2023. RemoteTouch: Enhancing Immersive 3D Video Communication with Hand Touch. In *VR 2023.* IEEE, 1–10.

[90] Zhengyou Zhang. 2000. A flexible new technique for camera calibration. *PAMI* 22, 11 (2000), 1330–1334.

[91] Jakob Zillner, Christoph Rhemann, Shahram Izadi, and Michael Haller. 2014. 3D-Board: A Whole-Body Remote Collaborative Whiteboard. In *UIST 2014.* ACM, 471–479.

[92] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan C. Russell, Max J. Argus, and Thomas Brox. 2019. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape From Single RGB Images. In *ICCV 2019.* IEEE, 813–822.

A APPENDIX

In the Appendix, we provide some details of the modules used in our network.

Fig. 10 illustrates the structure of the downsampling and upsampling blocks. The downsampling block commences with a 2D convolution layer featuring a kernel size of 3x3 and a stride of 2, which is then followed by batch normalization and a leaky ReLU activation. Conversely, the upsampling block performs the inverse process. Blocks at the same level are interconnected through skip connections, as depicted in Fig. 10.

Upon completing the downsampling process five times, we obtain a $4 \times 3 \times 512$ latent code that is subsequently passed through a
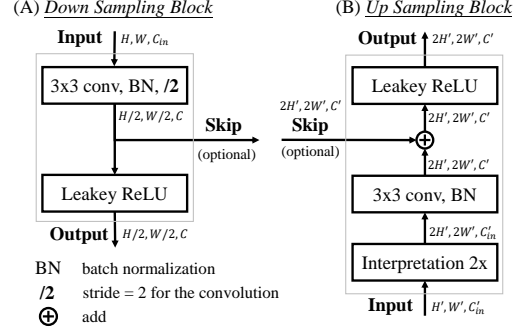

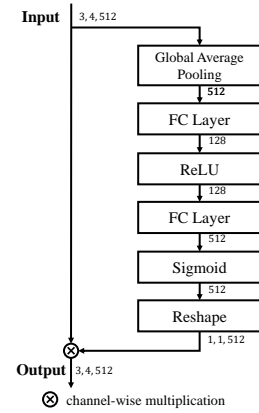**Figure 10: Our down sampling block and up sampling block.**

Figure 11: Illustration of the SE-block in our model.

Squeeze-and-Excitation (SE) block, as illustrated in Fig.11. Adhering to the standard implementation of the SE block as described in [24], this method has been demonstrated to enhance performance through our ablation study.