EgoPressure: A Dataset for Hand Pressure and Pose Estimation in Egocentric Vision

Yiming Zhao[1*] Taein Kwon[1*] Paul Streli[1*] Marc Pollefeys[1,2] Christian Holz[1]
[1]ETH Zürich [2]Microsoft Spatial AI Lab, Zürich
yiming-zhao.github.io/EgoPressure

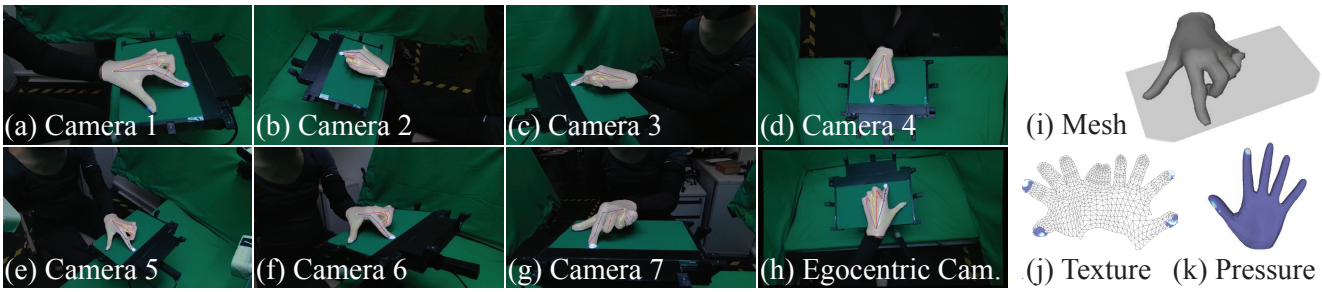


Figure 1. **The EgoPressure dataset.** We introduce a novel egocentric pressure dataset with hand poses. We label hand poses using our proposed optimization method across all static camera views (Cameras 1–7). The annotated hand mesh aligns well with the egocentric camera's view, indicating the high fidelity of our annotations. We project the pressure intensity and annotated hand mesh (Fig. *i*) to all camera views (Fig. *a* to *h*), and further provide the pressure applied over the hand as a UV texture map (Fig. *j* and *k*).

Abstract

Touch contact and pressure are essential for understanding how humans interact with objects and offer insights that benefit applications in mixed reality and robotics. Estimating these interactions from an egocentric camera perspective is challenging, largely due to the lack of comprehensive datasets that provide both hand poses and pressure annotations. In this paper, we present EgoPressure, an egocentric dataset that is annotated with high-resolution pressure intensities at contact points and precise hand pose meshes, obtained via our multi-view, sequence-based optimization method. We introduce baseline models for estimating applied pressure on external surfaces from RGB images, both with and without hand pose information, as well as a joint model for predicting hand pose and the pressure distribution across the hand mesh. Our experiments show that pressure and hand pose complement each other in understanding hand-object interactions.

1. Introduction

Understanding touch during hand-object interaction, especially from an egocentric perspective, is key for augmented reality (AR) [33, 81], virtual reality (VR) [22, 75], and robotic manipulation [9, 10, 55]. In AR/VR environments, touch contact and pressure information allow for more precise control and feedback [8]. For example, a virtual piano could vary its sound with key pressure, a feature lacking in current AR/VR systems [57]. Pressure sensing is also crucial for robots to replicate human grasping, where precise force estimation remains a challenge [9, 10, 45].

Previous approaches have used gloves [53, 54] and robots with tactile sensors [45, 90] to capture pressure measurements during object manipulation. However, this instrumentation interferes with natural touch by obstructing tactile feedback. In contrast, vision-based estimation methods require no instrumentation of the hands, and cameras are already integrated into devices like smart glasses and mixed reality headsets [23, 24]. Despite this potential, advancements in state-of-the-art models have been limited by the lack of relevant datasets with contact pressure annotations. A notable exception is the PressureVision dataset [21] that comprises RGB footage from four static cameras of hands interacting with a pressure-sensitive surface and corresponding projected pressure images.

In this paper, we introduce a novel dataset, EgoPressure, that extends these prior efforts [21, 22] and captures hand-surface interactions from an *egocentric* perspective, complete with pressure maps projected onto the articulated *hand mesh* in 3-space. Our capture platform combines a Sensel Morph touchpad with one head-mounted and seven static

* Equal contribution.

cameras, all recording RGB-D data at 30 Hz (Figure 1). The dataset includes 5 hours of footage from 21 participants, each performing 64 interaction sequences with an average length of 420 frames—making it the first bare-handed egocentric dataset with pressure and hand mesh annotations.

We further provide baseline models to demonstrate the potential of our dataset and establish a benchmark for future research. First, we set PressureVisionNet [21] as a baseline on our egocentric dataset and compare it to adapted models that incorporate hand pose as additional input. The model using hand poses estimated from the RGB images via the HaMeR [62] estimator outperforms PressureVisionNet by more than 5% in volumetric IoU error, with improvements of over 7% when using ground-truth hand poses. Additionally, we introduce the first model to jointly estimate hand pose, hand mesh, and pressure both over the mesh and on the surface from an egocentric RGB camera, thereby localizing contact and pressure in 3D space.

We summarize our key contributions as follows:

1. EgoPressure is the first egocentric hand-surface interaction dataset with projection-based pressure annotations together with 3D hand meshes.
2. We establish two novel benchmarks: (1) estimating contact pressure from egocentric RGB images with and without hand pose information, and (2) jointly predicting 3D hand poses and applied pressure, including the localization of pressure on a user's hand mesh.

EgoPressure thus offers new opportunities for future work to address the unique challenges of egocentric camera views and to precisely localize pressure on a user's hand.

2. Related Work

Vision-based hand-object pose estimation Over the past decade, significant progress has been made in hand tracking due to advancements in deep learning techniques [28, 60, 62] and the collection of relevant datasets [59, 85, 89]. While egocentric hand tracking for gesture recognition and direct input has advanced to the point of integration into modern commercial devices such as AR and VR headsets [29, 30], understanding hand interactions with external objects remains an active area of research [17, 18, 24, 44, 51]. Datasets gathered to aid machine understanding of such hand-object interactions rely on additional instrumentation of the users' hands [18], motion capture systems with hand-attached markers [17, 77], or multi-view camera rigs [5, 27, 44, 84, 86] to capture accurate ground-truth poses of users' hands under the higher degree of occlusion caused by the object.

Hand-object contact estimation In addition to object-relative hand pose, prior work has aimed to estimate contact points between the users' hands and external objects [17, 77]. Research has shown that when used as input proxies, real-world physical objects improve input control and provide haptic feedback [8]. For interactive research purposes, external tracking systems [8, 66] and wearable sensors such as acoustic sensors [70] and inertial measurement units were used to estimate contact [19, 25, 57, 71, 73, 78]. Additionally, vision-based techniques have been developed that use fiducial markers [46], active illumination for shadow creation [48, 81], vibration detection [74], or depth sensing [16, 26, 82, 83]. More recent work estimates touch using passive cameras without additional instrumentation on the user's hand or surface, enabling deployment on commercial mixed reality headsets [67, 75]. More detailed contact maps are inferred based on the intersection of tracked hand and object meshes [17, 20, 44, 64, 77, 86], requiring sub-millimeter accuracy—a challenging task for complex gestures due to soft tissue dynamics. To address this, Brahmbhatt et al. [3] used thermal imaging to obtain accurate contact maps. Additionally, prior efforts have utilized simulations to obtain more granular labels about contacting tissue [11, 31, 88].

Hand pressure estimation Moving beyond the mere detection of contact, prior work has estimated the pressure forces applied during hand interactions, which is crucial for robotic grasping tasks [9, 55] and provides an additional control dimension for input [65]. To estimate pressure from monocular images, visual cues such as fingernail alterations [6, 56] or surface deformations [36, 58] during press events have been used. Changes in object trajectory and interaction forces [15, 47, 63] also offer insights but are ineffective with static objects like tables and walls. Accurate pressure labels for training usually require instrumenting the user's hands with gloves [4, 49, 76] or the surface with force sensors [21, 63, 69], ideally flexible or conforming to various shapes [2, 42, 53]. However, this alters the visual appearance and tactile features of the hands and surface, affecting interaction and limiting generalization to bare hands and uninstrumented surfaces. Grady et al. [21, 22] collected two datasets with ground-truth pressure maps using a Sensel Morph [38] pressure sensor to train a neural network for estimating contact regions on surfaces from single RGB images. However, their method relies solely on exocentric static cameras that clearly capture the fingertips.

With EgoPressure, we provide the first dataset containing egocentric and multi-view RGB-D images of a bare hand interacting with a surface, along with synchronized pressure data, hand poses, and meshes (see Table 1).

3. Marker-less Annotation Method

To capture accurate hand poses and meshes without markers, we developed a multi-camera hand pose annotation method using the MANO hand model [68], differentiable rendering and multi-objective optimization. Figure 2 shows an overview of our method, which relies on C static cameras

Dataset	frames	participants	hand pose	hand mesh	markerless	real	egocentric	multiview	RGB	depth	contact	pressure	
												surface	hand
EgoPressure (ours)	4.3M	21	✓	✓	✓	✓	✓	✓	✓	✓	Pressure sensor	✓	✓
ContactLabelDB [22]	2.9M	51	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
PressureVisionDB [21]	3.0M	36	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
ContactPose [3]	3.0M	50	✓	✓	✓	✓	✗	✓	✓	✓	Thermal imprint	✗	✗
GRAB [77]	1.6M	10	✓	✓	✗	✓	✗	✗	✗	✗	Inferred from Pose	✗	✗
ARCTIC [17]	2.1M	10	✓	✓	✗	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
H2O [44]	571k	4	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
OakInk [84]	230k	12	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
OakInk-2 [86]	4.01M	9	✓	✓	✓	✓	✓	✓	✓	✗	Inferred from Pose	✗	✗
DexYCB [5]	582k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
HO-3D [27]	103k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
TACO [52]	5.2M	14	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗

Table 1. **Comparison between EgoPressure and selected hand-contact datasets.** The majority of prior datasets infer contacts based on hand and object pose. ContactLabelDB and PressureVisionDB also include ground-truth touch pressure but are limited to static cameras and do not provide accurate hand poses and meshes. Please see Supp. for the full table.

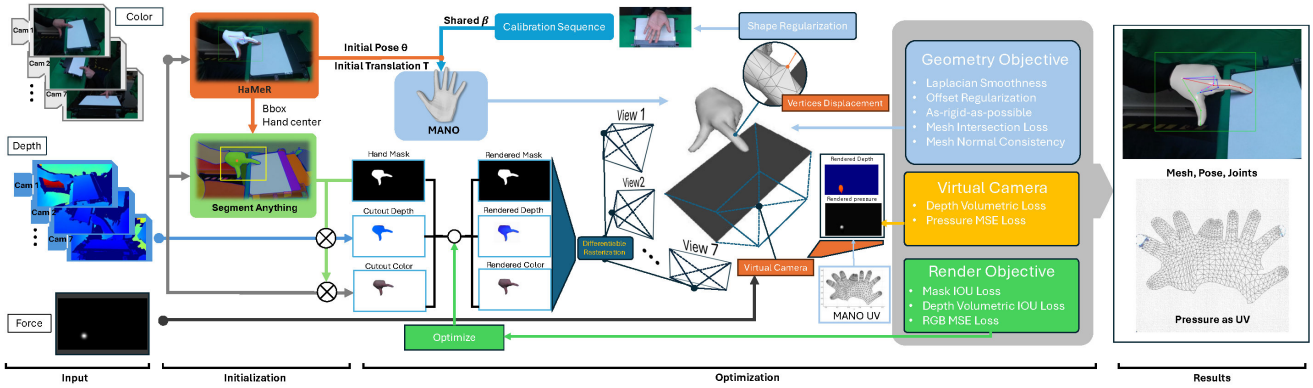


Figure 2. **Method overview.** The input for our annotation method consists of RGB-D images captured by 7 static Azure Kinect cameras and the pressure frame from a Sensel Morph touchpad. We leverage Segment-Anything [43] and HaMeR [62] to obtain initial hand poses and masks. We refine the initial hand pose and shape estimates through differentiable rasterization [7] optimization across all static camera views. Using an additional virtual orthogonal camera placed below the touchpad, we reproject the captured pressure frame onto the hand mesh by optimizing the pressure as a texture feature of the corresponding UV map, while ensuring contact between the touchpad and all contact vertices.

and a pressure-sensitive touchpad. Please see Supp. S2.2 for a detailed evaluation of our annotation method.

3.1. Automatic hand pose initialization

We use HaMeR [62] to estimate an initial MANO hand pose $\boldsymbol{\theta}_{\text{init}}$ and translation $\boldsymbol{t}_{\text{init}}$ for each static camera. Since HaMeR's prediction is based on a single RGB image, there is a scale-translation ambiguity, which we resolve by triangulating the root joints from the 7 static camera views. The orientation and hand pose are then initialized based on the output of a single camera view. HaMeR also provides a bounding box, which we use along with the 2D projected hand root as input to Segment-Anything (SAM) [43], from which we obtain an annotated segmentation mask $\boldsymbol{M}_{\text{gt}}$ for the hand in each static camera image.

3.2. Annotation refinement

Based on the initial hand pose, we obtain refined hand pose annotations via the following optimization using the input from the C *static* cameras. We use the MANO [32, 68] model for mesh representation with 25 PCA components and employ the DIB-R [7] differentiable renderer. The annotations include the hand pose $\boldsymbol{\theta}$, hand translation $\boldsymbol{t}$, vertex displacement $\boldsymbol{D}_{\text{vert}}$ in world coordinates, and the pressure over the hand mesh in the form of a texture map $\mathcal{T}_P$. All static cameras are pre-calibrated, allowing us to project the hand mesh into the frame of each static camera i using the extrinsic parameters $[\boldsymbol{R}^i_{\text{cam}}|\boldsymbol{t}^i_{\text{cam}}]$.

β-calibration For the MANO shape parameters β, we use separate calibration sequences for each hand of each participant, during which the participant slowly turns their hand to be visible from all cameras, with fingers spread. For these sequences, we also optimize the MANO shape parameters β with l_2 regularization in the previous optimization. The shape parameters are then reused across all other sequences for the participant, keeping β fixed during subsequent optimizations.

Following HARP [41], our method consists of two stages: (1) POSE OPTIMIZATION and (2) SHAPE REFINEMENT.

In the first stage, POSE OPTIMIZATION, we annotate the hand pose $\boldsymbol{\theta}$ and translation $\boldsymbol{t}$. The hand mesh $\boldsymbol{\Theta}$ can be derived directly from the MANO model [32], expressed

as $\mathbf{\Theta} = \text{MANO}(\boldsymbol{\theta}, \beta) + \boldsymbol{t}$. We note that certain parts of the hand, such as fingers, may not be visible from all camera angles—for instance, fingers obscured by the palm in a curled gesture. To address this, we incorporate the mesh intersection loss $\mathcal{L}_{\text{insec}}$ [40, 79]. The objective function is then defined as:

$$\mathcal{L}_{\text{pose}}(\mathbf{\Theta}) = \mathcal{L}_{\mathcal{R}}(\mathbf{\Theta}) + \mathcal{L}_{\text{insec}}(\mathbf{\Theta}). \quad (1)$$

The rendering objective $\mathcal{L}_{\mathcal{R}}$ and the mesh intersection loss $\mathcal{L}_{\text{insec}}$ will be detailed in Supp. S2.1.

In the SHAPE REFINEMENT stage, the pose $\boldsymbol{\theta}$ and translation $\boldsymbol{t}$ of the hand remain fixed. The optimization process introduces vertex displacement $\boldsymbol{D}_{\text{vert}}$. Each vertex is adjusted by an offset along its normal vector $\vec{\boldsymbol{n}}$, which is computed from the last epoch of the POSE OPTIMIZATION stage, to minimize the rendering loss $\mathcal{L}_{\mathcal{R}}(\mathbf{\Theta}^*)$. Consequently, the refined hand mesh $\mathbf{\Theta}^*$ can be expressed as $\mathbf{\Theta}^* = \mathbf{\Theta} + \vec{\boldsymbol{n}} \cdot \boldsymbol{D}_{\text{vert}}$. To ensure a reasonable mesh, the geometry objective $\mathcal{L}_{\mathcal{G}}$ is also included in the optimization. Additionally, we introduce a virtual render $\check{\mathcal{R}}$ to optimize pressure as a UV map $\mathcal{T}_P$ and minimize the distance between the hand mesh $\mathbf{\Theta}^*$ and the contact area on the surface of the touchpad. The objective function $\mathcal{L}_{\text{shape}}$ for this stage is as follows:

$$\mathcal{L}_{\text{shape}}(\mathbf{\Theta}^*) = \mathcal{L}_{\mathcal{R}}(\mathbf{\Theta}^*) + \mathcal{L}_{\mathcal{G}}(\mathbf{\Theta}^*) + \mathcal{L}_{\check{\mathcal{R}}}(\mathbf{\Theta}^*). \quad (2)$$

The virtual render $\check{\mathcal{R}}$, and its objective $\mathcal{L}_{\check{\mathcal{R}}}$ will be explained in the next section and the other terms in the geometry objective $\mathcal{L}_{\mathcal{G}}$ will be detailed in Supp. S2.1.2.

3.2.1. Virtual Render for Contact and Pressure

As shown in Figure 2, we also incorporate the captured pressure data in the optimization as a hand mesh texture feature for our proposed virtual rendering method. For this, we position a virtual orthogonal camera $\check{\mathcal{R}}$ under the touchpad, oriented upwards in the world coordinate system. The render size matches the resolution of the touchpad, and the camera's plane overlaps with the touchpad's sensing surface. The goal is for the rendered pressure $\check{\mathcal{R}}_P(\Theta^*, \mathcal{T}_P)$ on the hand mesh, with texture mapping of an optimized pressure UV map $\mathcal{T}_P$, to align with the input pressure $\boldsymbol{P}_{\text{gt}}$.

Additionally, we infer the contact area from $\boldsymbol{P}_{\text{gt}}$ using a simple pressure threshold. Using this contact area as a mask, we ensure that the masked rendered z-axis depth $\check{\mathcal{R}}_D(\Theta^*)[z]$ aligns with the distance Z_{v2p} from the camera to the touchpad, thereby ensuring physical contact.

The objective function $\mathcal{L}_{\check{\mathcal{R}}}(\mathbf{\Theta}^*)$ for the virtual render is:

$$\mathcal{L}_{\check{\mathcal{R}}}(\mathbf{\Theta}^*) = \text{MSE}(\check{\mathcal{R}}_P(\Theta^*, \mathcal{T}_P), \boldsymbol{P}_{\text{gt}}) + \left| \mathbb{I}(\boldsymbol{P}_{\text{gt}} > 0) \odot (\check{\mathcal{R}}_D(\Theta^*)[z] - Z_{v2p}) \right|_1. \quad (3)$$

4. EgoPressure Dataset

EgoPressure comprises 4.3M RGB-D frames (2560×1440 for static camera, 1920×1080 for egocentric camera) capturing interactions of both left and right hands (see Figure 7) with a touch and pressure-sensitive planar surface. The dataset features 21 participants performing 31 distinct gestures, such as touch, drag, pinch, and press, with each hand (see Figure 6). It includes a total of 5.0 hours of hand gesture footage comprised of synchronized RGB-D frames from seven calibrated static cameras and one head-mounted camera, along with ground-truth pressure maps from the pressure-sensitive surface captured at a frame rate of 30 fps. We used four different surface textures for the data capture rig, which also includes a green wall to facilitate synthetic background augmentation. Additionally, we provide high-fidelity hand pose and mesh data for the hands during interactions based on our proposed annotation method (see Section 3), as well as the tracked pose of the head-mounted camera. With EgoPressure, we offer a substantial dataset for egocentric hand pose and pressure estimation during interactions with rigid surfaces, thereby advancing machine understanding of human interaction with their surroundings through the fundamental modality of touch.

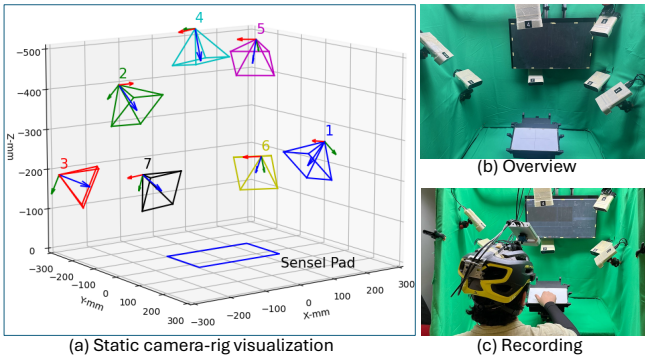


Figure 3. **7 static + 1 egocentric camera rig.**

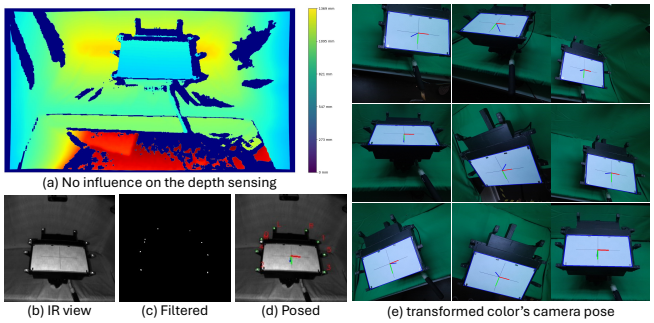


Figure 4. **Camera pose tracking with IR makers.**

4.1. Data capture setup

To capture accurate ground-truth labels for hand pose and pressure from egocentric views, we constructed a data capture rig that integrates a pressure-sensitive touchpad (Sensel Morph [38]) for touch and pressure sensing, along with

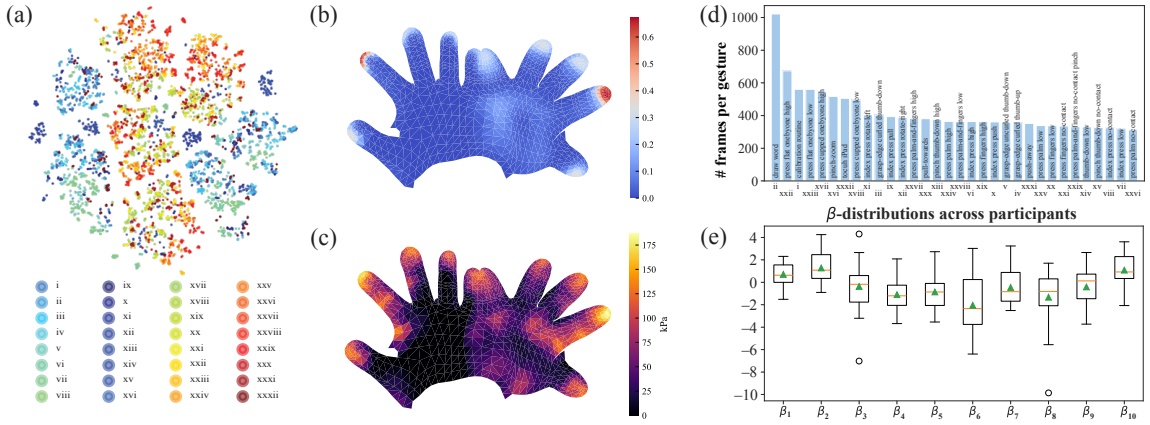
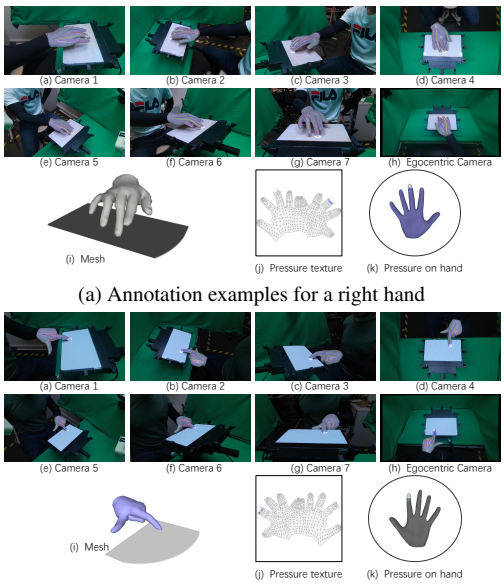


Figure 5. **Dataset Statistics (a)** t-SNE [80] visualization of hand pose frames θ over our dataset, with color coding for different gestures. All gestures are listed in Table S5 of the Supp. **(b)** Ratio of touch frames with contact for each vertex. **(c)** Maximum pressure over hand vertices across the dataset. **(d)** Mean length of performed gestures. **(e)** Distribution of β values across participants.

Figure 6. **Thumbnail of different poses in egocentric views.**



(a) Annotation examples for a right hand

(b) Annotation examples for a left hand

Figure 7. **Sample data from EgoPressure.**

seven static and one head-mounted RGB-D camera (Azure Kinect [1]) to capture RGB and depth images (see Figure 3). The touchpad (Sensel Morph), measuring 240 × 169.5 mm, is mounted on a tripod head. We use four different texture overlays (white, green, dark wood, light wood) printed on paper and placed over the Sensel Morph pad across participants. The seven static Azure Kinect cameras are attached to the aluminum frame, and the head-mounted camera is fixed on a helmet. The frame also holds a computer display and is surrounded by a green screen.

All cameras and the touchpad are connected to two workstations (Intel Core i7-9700K, Nvidia GeForce RTX 3070), their timestamps are synchronized via a Raspberry Pi CM4 using PTP, which also triggers all Azure Kinect cameras simultaneously at a frame rate of 30 fps. We varied the Kinect camera exposure (2.5 ms or 10 ms) and overhead lighting in three conditions across participants: dark (2 tubes active, 2.5 ms), medium (2 tubes, 10 ms), and bright (4 tubes, 10 ms). We minimize reliance on shadows via diffuse light sources.

Head-mounted camera tracking To obtain accurate poses of the head-mounted camera, we attach nine active infrared markers around the Sensel Morph pad in an asymmetric layout (see Figure 4). These markers, controlled by the Raspberry Pi CM4, are identifiable in the Azure Kinect's infrared image using simple thresholding (saturating the range of values of the infrared camera). The markers are turned on simultaneously, allowing for the computation of the camera pose via Perspective-N-Points and enabling an accurate evaluation of the temporal synchronization between cameras and the touchpad (see Supp. S3.5).

4.2. Participants

We recruited 21 participants from our institution (6 female, 15 male, ages 23–32 years, mean age = 26 years), ensuring a broad representation to cover broad hand anatomies. Participants' heights ranged from 160–194 cm (mean = 174, SD = 9), weights from 51–95 kg (mean = 69, SD = 14), and middle finger lengths from 7.3–9.2 cm (mean = 7.9, SD = 0.5) (see Figure 5 for distribution of MANO β-values).

4.3. Data acquisition procedure

Participants sat on an adjustable stool in front of the apparatus, wearing a helmet with a mounted camera pointing towards the Sensel Morph and a black arm sleeve on each arm up to the wrist. Before starting the data capture, the experimenter explained the task and the purpose of the study. They then signed a consent form and provided demographic information. The participants first performed a calibration gesture by slowly turning each hand, with fingers spread, within the camera rig. After calibration, participants conducted 31 different gestures, including touch, press, and drag gestures of varying strength, with each hand on the Sensel Morph touchpad (see Supp. S3.1 for a description of gestures). Each gesture was repeated 5 times if it involved a single touch action (e.g., press index finger) and 3 times if it involved a sequence of sequential touches (e.g., draw letters). Before each gesture, participants watched a video demonstrating how to perform the corresponding gesture with written instructions on a computer monitor in front of them. The experimenter guided the participants throughout the study, which took around 1 hour per participant. Participants could take a break after each gesture and received a chocolate bar as gratitude for their participation. In total, we recorded 6216 different gestures, i.e., 21 participants $\times$ 2 hands $\times$ (1 calibration + 27 $\times$ 5 + 4 $\times$ 3) gestures.

4.4. Data statistics

The average length of each motion sequence is 14 seconds, with an almost equal balance between frames capturing the left and right hands. Figure 5 shows the mean sequence lengths across gestures. Approximately 45.1% of all frames capture the hand in contact with the pressure-sensitive pad. Figure 5b visualizes the ratio of contact frames with a given vertex touching the surface, and Figure 5c shows the maximum pressure measured for each vertex. Following Grady et al. [21], we set a threshold of 0.5 kPa as the minimum effective pressure to discard diffuse readings from the touchpad.

5. Benchmark Evaluation

Previous work estimates applied pressure maps using only RGB images [21, 22]. With EgoPressure, we explore the advantages of incorporating accurate hand poses as additional input, which naturally provide richer context about the interaction. We introduce new benchmarks for estimating hand pressure using both RGB images and 3D hand poses. Additionally, we propose a novel network architecture that jointly estimates, from a single RGB image, the pressure applied to both an external surface and across the hand, providing a deeper understanding of the regions of the hand involved throughout the interaction.

5.1. Image-projected Pressure Baselines

We test our hypothesis that incorporating hand pose as an additional input enhances pressure estimation. To this end, we design a straightforward extension of PressureVisionNet [21]. Specifically, we augment the original encoder-decoder segmentation architecture, which was designed for RGB inputs only, by adding an additional channel for 2.5D hand keypoints. This involves projecting the 21 3D hand joints onto the image plane and incorporating their depth (z-coordinate) from the egocentric camera's coordinate system, scaled to millimeters.

We evaluate PressureVisionNet and the pose-augmented network on EgoPressure in three setups: (1) trained/tested on egocentric views, (2) trained/tested on the same exocentric views, and (3) trained on camera views 2,3,4,5; tested on 1,6,7. In all experiments, data from 15 participants is used for training and validation, while data from 6 participants is held out as the test set. To evaluate the augmented network, we use both the ground-truth hand joints from our annotations and the predicted hand joints from HaMeR [62]. The HaMeR-estimated hand poses serve as a fair baseline, reflecting the performance of state-of-the-art RGB-based hand pose estimators, while the ground-truth joints provide an upper bound, demonstrating the potential improvements achievable with more accurate hand poses.

The results are summarized in Table 2. Incorporating 2.5D hand joints improves performance in both egocentric and exocentric views and enhances generalization to unseen camera views. Figure 8 provides additional qualitative results, demonstrating the benefits of incorporating hand pose information. Further details on the architecture, training process, and evaluation metrics can be found in Supp. S1.1

Model	Train	Eval.	Modality	Cont. IoU ↑	Vol. IoU ↑	MAE ↓	Temp. Acc. ↑
PressureVisionNet [21]	ego.	ego.	RGB	55.73	38.64	53.60	91.68
[21] w. [62] pose	ego.	ego.	RGB & pred pose	56.25	40.52	55.23	91.67
[21] w. GT pose	ego.	ego.	RGB & GT pose	58.80	41.39	53.79	92.17
PressureVisionNet [21]	exo. (2,3,4,5)	exo. (2,3,4,5)	RGB	62.11	44.73	43.15	93.61
[21] w. [62] pose	exo. (2,3,4,5)	exo. (2,3,4,5)	RGB & pred pose	62.95	45.01	42.53	93.83
[21] w. GT pose	exo. (2,3,4,5)	exo. (2,3,4,5)	RGB & GT pose	64.39	47.58	41.72	94.18
PressureVisionNet [21]	exo. (2,3,4,5)	exo. (1,6,7)	RGB	36.82	25.05	62.22	83.40
[21] w. [62] pose	exo. (2,3,4,5)	exo. (1,6,7)	RGB & pred pose	38.46	28.10	51.50	86.34
[21] w. GT pose	exo. (2,3,4,5)	exo. (1,6,7)	RGB & GT pose	43.04	31.39	49.45	89.78

Table 2. **Image-projected pressure estimation using different inputs.** Our high-fidelity hand pose annotations improve contact IoU [%], volumetric IoU [%], MAE [Pa], and temporal accuracy [%] compared to using no hand poses or HaMeR [62] hand poses as additional input for novel exocentric and egocentric views.

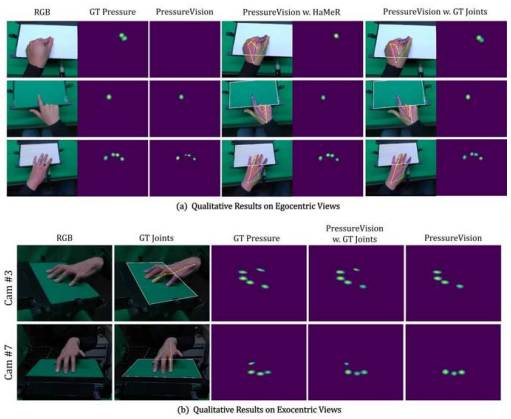


Figure 8. **Qualitative results**. We present the egocentric experiment results in Subfigure (a). In Subfigure (b), both baseline models are trained using camera views 2, 3, 4, and 5. We display the results for one seen view and one unseen view. Additionally, we overlay the 2D keypoints predicted by HaMeR [62] and our annotated ground truth on the input image. For better visualization, the contour of the touch sensing area is also highlighted as a reference.

5.2. First Hand-Projected Pressure Baseline

Both the original PressureVision framework [21] and its subsequent iteration, PressureVision++ [22], predict 2D hand pressure on the image plane. However, this introduces ambiguity about the exact manifestation of this pressure between hands and objects within the 3D space.

To address this, we introduce a new baseline model, *PressureFormer*, which estimates pressure as a UV map of a 3D hand mesh, enabling projection both as 3D pressure onto the hand surface and as 2D pressure onto the image space.

As illustrated in Figure 9, our model builds upon HaMeR [62]. It processes the hand vertices V_{hand} in the camera frame and the image feature tokens from HaMeR's Vision Transformer (ViT) [14]. A transformer-based decoder receives V_{hand} as multiple input tokens while cross-attending to the image feature tokens from the ViT. Each output token represents a D-dimensional feature for a corresponding mesh vertex, which we then map onto a UV feature map using the UV coordinates of the MANO model [68]. Given the sparsity of the UV feature map post-projection, we apply two convolutional layers for neural interpolation and reduce the dimensions to the number of force classes C to predict the quantified UV-pressure map U_{pred}.

Firstly, we compute the coarse UV-pressure loss $\mathcal{L}_c$ between U_{pred} and the ground-truth UV-pressure map U_{gt}, quantified from the scalar UV-Pressure $\mathcal{T}_P$ of our dataset. Subsequently, we render the pressure P_{pred} back onto the original image plane using the M_{hand} mesh of vertices V_{hand} and texture mapping the predicted U_{pred} UV-pressure map. Using a differentiable renderer [7], we invert the z-normal and z-axis of the face vertices to identify the mesh faces furthest from the camera (i.e., occluded vertices) as places of contact. This allows us to compute the pressure loss $\mathcal{L}_p$ against the ground-truth pressure P_{gt}. Both $\mathcal{L}_p$ and $\mathcal{L}_c$ employ cross entropy loss. The training of PressureFormer is supervised by a loss function defined as:

$$\mathcal{L}_{\mathcal{PF}} = w_1\mathcal{L}_c + w_2\mathcal{L}_p. \tag{4}$$

For comparison, we project the image-based pressure maps from PressureVisionNet and its hand-pose-augmented baseline onto the corresponding hand mesh estimated from the same image using HaMeR [62]. Similarly, this process involves identifying the hand mesh faces farthest from the camera and rasterizing the 2D pressure map onto the UV map (see Supp. Figure S2). We also evaluate a variant of PressureFormer trained without explicit UV loss supervision. We thus introduce a benchmarking task that assesses the accuracy of pressure estimation on the hand surface and the performance of jointly estimating pressure and hand mesh.

We trained our PressureFormer and baseline models using images from all camera views of 15 participants, incorporating a hand-centered crop. During training, we applied data augmentation techniques, including shifting, rescaling, and rotating. We evaluated the models on a held-out test set of six participants using (1) all camera views and (2) only egocentric camera images. Additionally, we assessed the generalization of the models to the test set of PressureVision [21]. We evaluate the accuracy of the estimated pressure map in both image space and UV space when ground truth data is available (see Supp. S1.3).

Results The results are summarized in Table 3. PressureFormer outperforms all image-projected pressure baselines in Contact IoU and Volumetric IoU on the UV pressure map. It also attains the highest Contact IoU on the image-projected pressure map and shows better generalization to PressureVision. The hand-pose-augmented baseline, which directly predicts pressure onto the camera image, achieves the best Volumetric IoU on the image-based pressure map. These results highlight the value of incorporating hand pose information for pressure estimation and jointly estimating hand pose and pressure for more coherent interaction modeling. Additionally, the results underline the value of the coarse UV-pressure loss in enhancing the accuracy of the pressure predictions on the UV map (see Supp. Figure S3). Figure 11 provides a qualitative comparison of the UV pressure maps estimated by the three baseline methods.

Generalization of PressureFormer PressureFormer employs a UV-pressure map that enhances the generalization of hand contact and pressure prediction for more complex objects. Unlike estimating pressure on the image plane, which focuses on hand-surface interactions, the UV map captures pressure on the hand vertices in 3D space. As PressureFormer uses the pretrained HaMeR [62] model as its backbone to extract hand vertices and image features from vision transformer tokens, it can effectively handle diverse

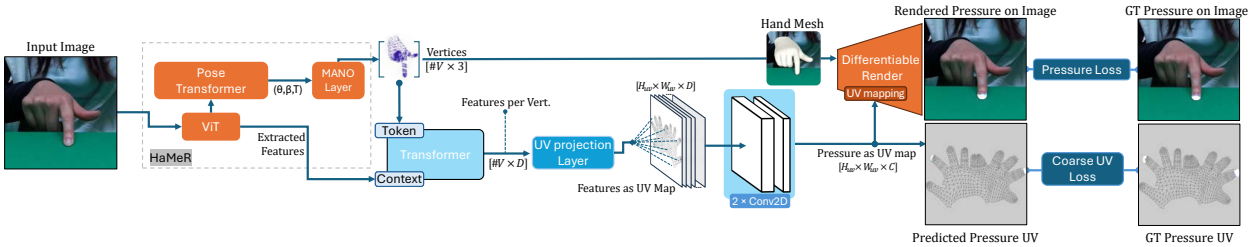


Figure 9. **PressureFormer** uses HaMeR's hand vertices and image feature tokens to estimate the pressure distribution over the UV map. We employ a differentiable renderer [7] to project the pressure back onto the image plane by texture-mapping it onto the predicted hand mesh.

Model	Eval. Dataset	Im. Contact IoU↑	Im. Vol. IoU↑	Im. MAE↓	Temp. Acc.↑	UV Press. Contact IoU↑	UV Press. Vol. IoU↑
PressureVisionNet [21]	EgoPressure (ego. & exo.)	40.71	32.11	**44**	90	21.53	16.41
[21] (w/ HaMeR [62])	EgoPressure (ego. & exo.)	42.52	**35.40**	49	**92**	24.10	17.36
PressureFormer (Ours)	EgoPressure (ego. & exo.)	**43.04**	31.57	71	89	**33.12**	**24.54**
PressureFormer (w/o $\mathcal{L}_c$)	EgoPressure (ego. & exo.)	41.27	29.57	74	88	26.24	18.61
PressureVisionNet [21]	EgoPressure (ego.)	40.65	**33.91**	**47**	**87**	26.59	19.81
PressureFormer (Ours)	EgoPressure (ego.)	**42.75**	30.57	89	83	**33.51**	**23.01**
PressureVisionNet [21]	PressureVision (exo.)	7.54	7.11	146	55	-	-
PressureFormer (Ours)	PressureVision (exo.)	**29.03**	**21.71**	**121**	**79**	-	-

Table 3. **Performance comparison** of our PressureFormer model against image-projected pressure baselines, evaluated using temporal accuracy [%], image-based pressure metrics (Image Contact IoU, Image Vol. IoU, Image MAE [kPa]), and UV map-based pressure metrics (UV Pressure IoU, UV Pressure Vol. IoU). PressureFormer demonstrates superior performance in UV pressure IoU and UV Pressure Vol. IoU, while also achieving higher scores in image-based Contact IoU. By directly predicting pressure on the UV map, PressureFormer offers advantages, enabling accurate 3D pressure reconstruction by projecting the results onto the estimated hand surface.

hand poses while integrating hand-centric image texture information. We provide qualitative results demonstrating PressureFormer's ability to generalize to unseen camera configurations, such as the integrated passthrough sensors of the Quest 3 (see Figure 10), as well as to unseen real-world objects and environments (see Supp. Figure S5).

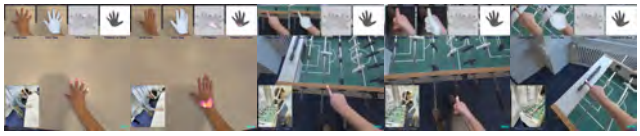


Figure 10. **PressureFormer** on data captured with Meta Quest 3.

6. Conclusion

Limitations Despite the promising generalization of the PressureFormer model, the EgoPressure dataset is limited to hand interactions with flat surfaces, as capturing precise pressure measurements on general objects without instrumenting the user's hands remains challenging. Additionally, the dataset was collected indoors and consists only of single-hand interactions. A natural extension would be to incorporate dual-hand scenarios in more diverse environments. For a detailed discussion of these limitations, see Supp. S4).

Summary In this paper, we introduce EgoPressure, a novel egocentric hand pressure dataset paired with a multi-view hand pose estimation and pressure annotation method. EgoPressure includes precise 3D hand meshes, multi-view RGB and depth images, egocentric view images, and pressure intensities. We establish a new benchmark and demonstrate the effectiveness of using hand pose data in pressure estimation. Furthermore, we introduce PressureFormer, a model that directly predicts pressure on the hand mesh, along with

Figure 11. **Qualitative Results PressureFormer on our dataset.** We compare our PressureFormer with both PressureVisionNet [21] and our extended baseline model with HaMeR-estimated [62] 2.5D joint positions. Additionally, we provide visualizations of the hand mesh estimated by HaMeR, alongside the 3D pressure distribution on the hand surface derived from our predicted UV-pressure in the last two columns. Note that we transform the left-hand UV maps into the right-hand format.

relevant baselines for comparison. In conclusion, we believe EgoPressure represents an important step toward enabling machines to better understand hand-object interactions by capturing 3D pressure from an egocentric view.

7. Acknowledgment

We greatly appreciate all the participants who voluntarily contributed to dataset collection. We also thank Zihan Zhu, Boyang Sun, and Shaohui Liu for their insightful discussions.

EgoPressure: A Dataset for Hand Pressure and Pose Estimation in Egocentric Vision

Supplementary Material

8. Details about Benchmark Evaluation

In this section, we provide further details about the benchmark evaluation experiments from Section 5.

8.1. Details for image-projected pressure baselines

8.1.1. Baseline model with Additional Keypoint depth maps

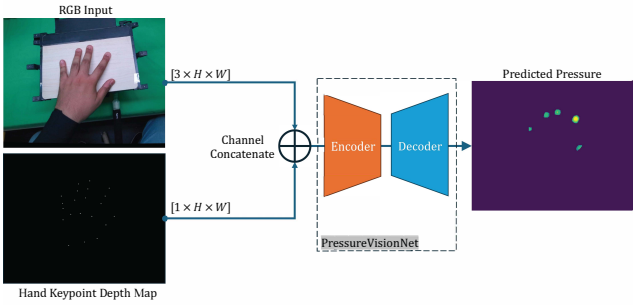


Figure 12. **Overview of the image-projected pressure baseline with additional hand pose input.** The baseline receives an RGB image and a keypoint depth map as inputs to an encoder-decoder segmentation network for pressure estimation.

The previous method [21] for predicting hand pressure relies solely on RGB images as inputs. In contrast, our new benchmark is designed to incorporate an additional modality, hand pose. To ensure a fair comparison between the baselines and our approach, we extend the existing method with additional hand pose inputs. In addition to the three RGB channels of PressureVision, we add a keypoint depth map as an additional input channel to the segmentation network.

Encoder-decoder segmentation network architecture. Similar to PressureVision, we employ an ImageNet-pretrained Squeeze-and-Excitation Network (SEResNeXt50) [34, 35] as the encoder, which takes both RGB and 3D hand pose inputs, and a feature pyramid network [37, 50] as the decoder, which generates a pressure map.

Training. For training, we use the Adam optimizer with a batch size of 8. The training process begins with a learning rate of 0.001 for 100k iterations, followed by 500k iterations with a learning rate of 0.0001.

8.1.2. Evaluation Metrics

For evaluation, we adopt the four metrics proposed in PressureVision [21]: Contact Intersection over Union (IoU), Volumetric IoU, Mean Absolute Error (MAE), and Temporal Accuracy.

Contact IoU measures the accuracy of contact surface predictions by calculating the IoU between the estimated and ground-truth binarized pressure maps. Volumetric IoU extends this by incorporating the accuracy of the predicted pressure magnitudes, calculated as the ratio of the sum of the minimum pressure values between the estimated and ground-truth pressure maps at each pixel to the sum of the maximum values. MAE quantifies the pressure prediction error in kilopascals (kPa) per pixel. Temporal Accuracy assesses the consistency of contact over time by verifying frame-by-frame contact consistency between the estimated and ground-truth values.

8.2. Additional Qualitative Results

More qualitative results for the baselines are provided in Figure 29. More qualitative examples for the annotations are shown in Figures 32, 33, 34, 35, 36 and 37.

We also present qualitative results from the third-person view camera experiments (refer to Table 2 in the main paper). Figure 30 and 31 include visual comparisons between our model, which uses RGB and a hand keypoint depth map, and PressureVisionNet [21] which uses only RGB input. Figure 30 shows the models' qualitative performance on images from cameras *2, 3, 4, and 5*, with both models trained on a separate training set from these views. In Figure 30, we evaluate the same models on novel views from cameras *1, 6, and 7*, which were not included in the training set.

In the second column of Figure 30 and Figure 31, the reprojected touch sensing area is shown as a white outline to verify the camera pose. We also provide MAE and Contact IoU values for each sample. Notably, including additional hand pose information enhances the model's ability to estimate pressure and contact, especially for occluded hand parts (see examples *04* in Figure 30 and *09, 11, 13* in Figure 31).

8.3. Additional Evaluation of PressureFormer

PressureFormer improves upon the baselines from Section 5.1 by estimating pressure directly on the UV map of the reconstructed hand mesh. This approach extends the representation of pressure via the estimated 3D hand pose into 3D space. While the hand mesh-based pressure representation can still be projected onto the image plane for benchmarking with prior methods [21, 22], it offers additional insights about the specific hand regions applying pressure. This capability is beneficial for scenarios involving

$$\mathcal{L}_{\mathcal{R}}(\boldsymbol{\Theta}) = \sum_{i=0}^{C} [\lambda_M \underbrace{(1 - \text{IoU}(\mathcal{R}_M^i(\boldsymbol{\Theta}), \boldsymbol{M}_{\text{gt}}^i))}_{\textbf{Mask IoU Loss } \mathcal{L}_M(\boldsymbol{\Theta})} + \lambda_A \underbrace{\text{MSE}(\mathcal{R}_F^i(\boldsymbol{\Theta}, \boldsymbol{\mathcal{T}}), \boldsymbol{I}_{\text{gt}}^i)}_{\textbf{Appearance Loss } \mathcal{L}_A(\boldsymbol{\Theta})} + \lambda_D \underbrace{(1 - \frac{|\min(\mathcal{R}_D^i(\boldsymbol{\Theta}), \boldsymbol{D}_{\text{gt}}^i)|_1}{|\max(\mathcal{R}_D^i(\boldsymbol{\Theta}), \boldsymbol{D}_{\text{gt}}^i)|_1})}_{\textbf{Depth Volumetric IoU Loss } \mathcal{L}_D(\boldsymbol{\Theta})}]. \quad (5)$$

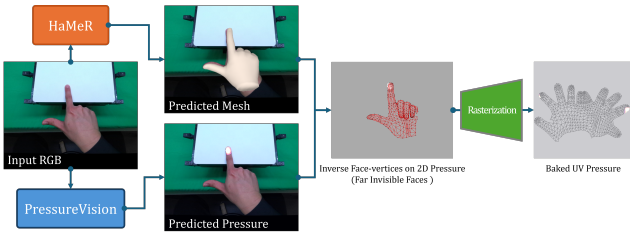


Figure 13. **Pipeline for projecting the image-based pressure map (from PressureVision) onto the UV map:** Starting with the predicted hand mesh and 2D pressure map, the normals and z-axis are inverted to identify occluded (invisible) faces of the mesh. The pressure is then mapped onto the UV space using rasterization.

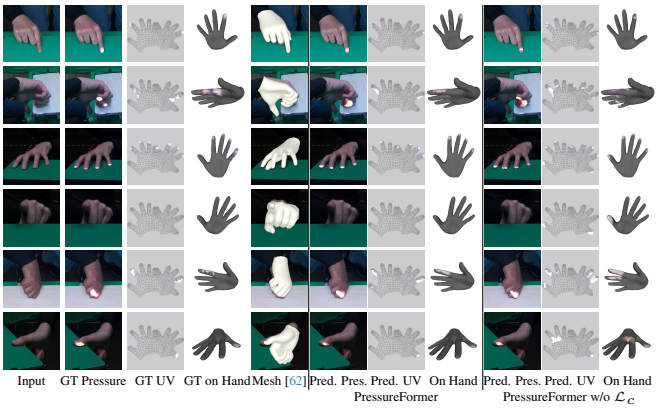


Figure 14. **Qualitative examples demonstrating the impact of coarse UV loss supervision $\mathcal{L}_c$.** The coarse UV loss supervision $\mathcal{L}_c$ prevents the prediction of pressure in areas of the UV map that are not rendered on the image plane (see Section 5.2). These regions typically correspond to faces oriented toward the camera, where pressure and contact are not physically possible.

complex hand-object interactions, such as when fingers are partially occluded or interacting with non-planar surfaces, where an image-projected pressure map may have limitations and introduce additional ambiguities. These tactile hand dynamics are also helpful for enabling precise grasping and object manipulation in humanoid robotics.

Evaluation Metrics. In Section 5.2, we compare PressureFormer with PressureVisionNet [21] and its hand keypoint depth map-augmented baseline, both of which directly estimate camera image-projected pressure maps. We make these comparisons based on the evaluation metrics established in PressureVision (see Table 2). We extend this evaluation by considering the hand mesh-projected pressure that PressureFormer directly estimates as a UV pressure map (see Figure 9).

To assess the accuracy of pressure estimation across the hand surface, we compute two metrics on the UV pressure map: Contact IoU and Volumetric IoU.

Training. During preprocessing, the images are cropped with a margin around the hand and resized to match the network's input dimensions. For evaluation, we ensure the hand remains centrally positioned in the frame throughout the cropping process. Data augmentation, including shifts, rescaling, and rotations, is applied across all methods. Training employs the Adam optimizer with a batch size of 8, using a learning rate of 0.001 for 100k iterations and 0.0001 for the subsequent 500k iterations. The loss function for PressureFormer (see Eq. 4) uses weighting parameters $w_1 = 0.2$ and $w_2 = 0.05$.

9. Details and Evaluation of Annotation Method

9.1. Optimization Objectives

In this section, we describe the optimization objectives necessary for complete implementation in conjunction with the objectives described in the main paper.

9.1.1. Render Objective

Since hand mesh $\boldsymbol{\Theta}$ is the only rendered object across all camera views, we use pseudo groundtruth mask $\boldsymbol{M}_{\text{gt}}^i$ from Segment-Anything (SAM) [43] to extract relevant regions, appearance $\boldsymbol{I}_{\text{gt}}^i = \boldsymbol{I}_{\text{in}}^i \otimes \boldsymbol{M}_{\text{gt}}^i$ and depth $\boldsymbol{D}_{\text{gt}}^i = \boldsymbol{D}_{\text{in}}^i \otimes \boldsymbol{M}_{\text{gt}}^i$, from input RGB image $\boldsymbol{I}_{\text{in}}^i$ and depth $\boldsymbol{D}_{\text{in}}^i$. For the optimization of the rendered appearance $\mathcal{R}_F^i(\boldsymbol{\Theta}, \boldsymbol{\mathcal{T}})$, a single texture is shared across all camera views within an input batch of several consecutive frames, which ensures that the mesh $\boldsymbol{\Theta}$ remains consistent across different cameras and consecutive frames. The rendering loss $\mathcal{L}_{\mathcal{R}}$ across all C cameras is represented in Eq. 5

Depth Volumetric IoU $\mathcal{L}_D(\boldsymbol{\Theta})$ [21] is defined in the third term of Equation 5. We apply it to the ground truth and rendered depth. In Table 4, we show these two losses: **Depth Volumetric IoU Loss** $\mathcal{L}_D(\boldsymbol{\Theta})$ and **Mask IoU Loss** $\mathcal{L}_M(\boldsymbol{\Theta})$ on the mesh from the initial input, i.e., $\boldsymbol{\theta}_{ini}$ and $\boldsymbol{t}_{ini}$, and two consecutive annotation stages, POSE OPTIMIZATION and SHAPE REFINEMENT.

Figure 15. **Qualitative comparison of UV Pressure.** We compare our PressureFormer model against the original PressureVision [21] and its extended version with additional hand keypoint inputs. For both PressureVision-based approaches, the UV pressure is obtained by baking the image-based pressure predictions onto the UV map of the hand mesh, using the hand mesh estimates provided by HaMeR [62].

Category	$\mathcal{L}_D \downarrow$			$\mathcal{L}_M \downarrow$		
	Initial	**Pose.**	**Pose. + Shape.**	**Initial**	**Pose.**	**Pose. + Shape.**
Overall	0.4443	0.1759	0.1317	0.3887	0.1165	0.0558
With Contact	0.4444	0.1752	0.1309	0.3891	0.1167	0.0562
Without Contact	0.4441	0.1790	0.1351	0.3871	0.1158	0.0545

Table 4. **Losses by Stages.** We validate the quality of hand poses using two metrics, Depth Volumetric IoU Loss $\mathcal{L}_D$ (Eq. 5) and Mask IoU Loss $\mathcal{L}_M$ (Eq. 5), computed on 386,231 $\times$7 (static cameras) = 2,703,617 annotated frames. Of these, 2,192,633 (81%) show the hand in contact with the touchpad. We report the results before (initial) and after each consecutive optimization step: POSE OPTIMIZATION and SHAPE REFINEMENT.

9.1.2. Geometry Objective

The geometry objective ($\mathcal{L}_\mathcal{G}$) is composed of several terms:

$$\mathcal{L}_\mathcal{G} = \mathcal{L}_{\text{insec}} + \mathcal{L}_{\text{arap}} + \mathcal{L}_{\vec{\mathbf{n}}} + \mathcal{L}_{\text{lap}} + \mathcal{L}_{\text{offset}} \tag{6}$$

The term $\mathcal{L}_{\text{insec}}$ represents the mesh intersection loss, which utilizes a BVH tree to identify self-intersections within the mesh. Penalties are subsequently applied based on these detections [40, 79].

The term $\mathcal{L}_{\text{arap}}$, as-rigid-as-possible loss, as introduced in [72], promotes increased rigidity in the 3D mesh while distributing length alterations across multiple edges. The variation in edge length is determined relative to the mesh from the last epoch of POSE OPTIMIZATION as

$$\mathcal{L}_{\text{arap}} = \frac{1}{|E|} \sum_{v^* \in \boldsymbol{\Theta}^*} \sum_{\boldsymbol{e}^* \in E(v^*, u^*)} |\|\boldsymbol{e}^*\| - \|\boldsymbol{e}^p\||, \tag{7}$$

where $E(v^*, u^*)$ is the edge connecting vertex v^* and u^* in the set of all edges E, and the edge $\boldsymbol{e}^p$ is formed by the corresponding vertices v^p and u^p in the mesh without vertex displacement $\boldsymbol{D}_{\text{vert}}$.

The mesh vertices $\mathbf{V}_{\boldsymbol{\Theta}^*}$ are smoothed by the Laplacian mesh regularization $\mathcal{L}_{\text{lap}}$ [13], and the normal consistency regularization $\mathcal{L}_{\vec{\mathbf{n}}}$ smooths normals on the displaced mesh. Finally, the vertex offset term $\mathcal{L}_{\text{offset}}$ is calculated by $\|\boldsymbol{D}_{\text{vert}}\|^2$.

9.1.3. Depth Culling

In some sequences, hands may be partially occluded by the Sensel Morph touchpad from certain camera views, which can hinder the convergence of the optimization process for the total rendered mask. To address this issue, we have modeled the touchpad and its pedestal. We pre-generate the depth map D_o to represent these scene obstacles. Subsequently, we perform simple depth culling with the rendered depth $\mathcal{R}_D$ by generating a culling mask $M_{dc} = \mathbb{I}(D_o > \mathcal{R}_D)$. This allows us to create cutouts on the rendered depth $\mathcal{R}_D$, the appearance $\mathcal{R}_F$, and the mask $\mathcal{R}_M$, which together represent the hand parts in front of the scene obstacles. After initial tests, we noticed that this depth culling encourages

Figure 16. **Qualitative evaluation of PressureFormer on diverse, real-world examples featuring various objects and scenes.** Despite being trained exclusively on EgoPressure, the model recognizes pressure regions during corresponding contact events, demonstrating its potential for generalization.

the intersection of the hand mesh and the touchpad to reach lower mask IoU loss $\mathcal{L}_M$. Therefore, we add a collision box of the touchpad into mesh intersection loss $\mathcal{L}_{insec}$ to penalize this intersection. We show an example in Figure 17.

9.1.4. Temporal Continuity

Our optimization considers consecutive captures consisting of 7 RGB-D and one pressure frame in batches of size B to ensure temporal continuity of annotated hand poses across timestamps. We apply regularization on the approximated second-order derivative of the hand joint positions $\mathbf{J}$, which are regressed from the MANO mesh. The temporal continuity regularization is:

$$\mathcal{L}_{\text{temp}} = \frac{1}{B-2} \sum_{i=1}^{B-2} \|\mathbf{J}_{i+2} - 2\mathbf{J}_{i+1} + \mathbf{J}_i\|_2 \,. \tag{8}$$

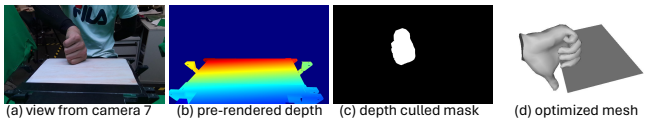


Figure 17. **Depth Culling.** (a) In the view of Camera 7, the thumb is behind the touchpad. (b) We compare the rendered depth of hand $\mathcal{R}_D$ and pre-rendered depth map of scene obstacles D_o, and (c) cutout the part which has a larger depth value than D_o. The thumb rendered in blue color is cutout due to the depth culling. (d) The collision box is rendered in 3D.

Losses	Ours	w/o $\mathcal{L}_A$	w/o $\mathcal{L}_D$	w/o $\mathcal{L}_{insec}$	w/o $\mathcal{L}_{arap}$	w/o $\mathcal{L}_{lap}$	w/o $\mathcal{L}_{\vec{n}}$	w/o $\mathcal{L}_{offset}$
$\mathcal{L}_D \downarrow$	**0.1251**	0.1404	0.1797	0.1368	0.1347	0.1396	0.1349	0.1477
$\mathcal{L}_M \downarrow$	**0.0488**	0.0662	0.0724	0.0619	0.0597	0.0654	0.0653	0.0728
3D tips error [mm] $\downarrow$	**5.68**	7.66	8.45	7.99	8.61	8.49	8.39	8.28

Table 5. **Quantitative evaluation of our annotation method compared to 3D tip positions triangulated from manual annotations.** We conduct an ablation study for the different loss terms, including appearance loss $\mathcal{L}_A$ (Eq. 5), depth volumetric IoU loss $\mathcal{L}_D$ (Eq. 5), mesh intersection loss $\mathcal{L}_{insec}$ (Sec. 9.1.2), as-rigid-as-possible loss $\mathcal{L}_{arap}$ (Sec. 9.1.2), Laplacian smoothness $\mathcal{L}_{lap}$ (Sec. 9.1.2), normal consistency regularization $\mathcal{L}_{\vec{n}}$ (Sec. 9.1.2), and vertex offset regularization $\mathcal{L}_{offset}$ (Sec. 9.1.2). We demonstrate that each loss term contributes to our optimization performance.

9.2. Evaluation of Annotation Fidelity

9.2.1. Manual Annotation and Inspection

To verify the quality of the hand poses from our annotation method, we manually annotated 300 randomly selected sets of 7 static views and one egocentric view (300×8 = 2400 frames). We annotated all the **visible** nail tips in the camera views, resulting in **7176** 2D points. These 2D nail tips were then triangulated to obtain 3D points. After applying a threshold of 2 pixels on the re-projection error to exclude inconsistent manual annotations, we obtained **1114** 3D points that were visible in at least two camera views. In Table 5, we report the distance error of the hand tips obtained from our annotation method relative to the 3D tip positions based on the manual annotations. We also include an ablation study of our approach. Qualitative results of the manual annotations and our method are shown in Figure 18.

9.2.2. Comparison to learning-based model

Compared to the state-of-the-art 3D hand pose estimator, HaMeR, our optimization-based method offers significant advantages, enabling the creation of high-quality annotations for our dataset. As shown in Figure 20, although hand poses from HaMeR [62] appear plausible from a top view, side

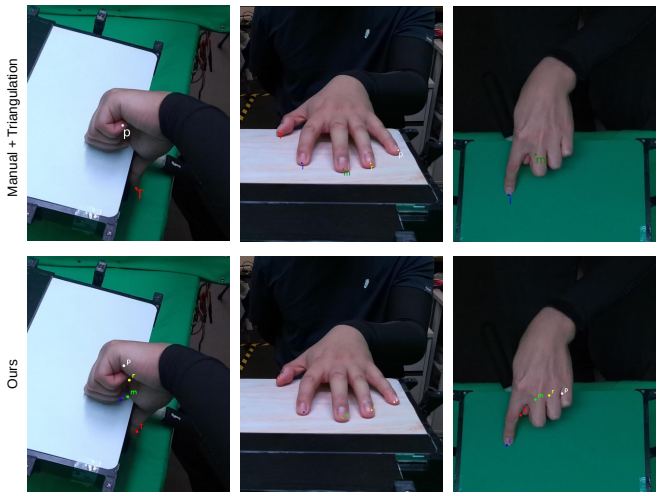


Figure 18. **Manual Verification Examples**. We demonstrate our annotation is accurate compared to the manual annotations. **(above)** We re-project the triangulated nail tips. We only triangulated them when they are visible in at least 2 views. **(bottom)** We re-project our 3D annotations which also show invisible nail tips as well.

views expose inaccuracies and scale ambiguities. In contrast, our annotation method produces robust and consistent results across all camera views. In Table 2, we demonstrate that the baseline model with our high-quality 3D hand poses improves hand pressure estimation compared to using HaMeR's [62] predictions.

To further evaluate annotation quality, we provide the validation results comparing the triangulation of predicted nail tips with manual annotations across static views in Table 6. Additionally, Figure 21 presents a qualitative comparison of pressure estimation incorporating additional poses from HaMeR [62] and our ground truth annotations. The results emphasize the importance of the high-fidelity hand pose annotations from our optimization method, both quantitatively and qualitatively, and highlight the necessity of advancing hand pose and pressure map estimation in future research.

Finally, we report the results of the HaMeR method after fine-tuning on our dataset in Table 7 and in Figure 19. Although fine-tuning improves performance, there remains room for further enhancement. These results establish a solid baseline for tackling 3D hand pose estimation during hand-surface interactions in an egocentric view.

	3D tips error [mm]	Std.
Ours	5.68	4.9
HaMeR [62]	12.37	6.3

Table 6. **Hand pose verification.** Triangulation is performed on the nail tips using HaMeR [62] predictions across all static cameras, compared against manual annotations.

	MPJPE [mm]	Reconstruction Error [mm]
Finetuned HaMeR [62]	10.75	6.10
HaMeR [62]	18.58	8.11

Table 7. **Fine-tuning results** of HaMeR [62] on EgoPressure demonstrate improved hand pose accuracy, underscoring the value of our dataset for 3D hand pose estimation.

Figure 19. **Hand pose prediction and ground truth pose visualization for each camera.** We fine-tune HaMeR [62] on our dataset, demonstrating improved detail in hand pose estimation, particularly in scenarios where the hand interacts with a surface.

10. Extended Details about Dataset

10.1. Details about Gesture Description

Table 8 lists all gestures performed by a participant during the data collection, including which hands were used and how often each gesture was repeated. We refer to the accompanying video for visual examples.

10.2. Details about Dataset File Format

For each timestamp, we provide a set of camera frames (Static Cameras 1 to 7 with a resolution of 2560×1440 and Egocentric Camera of 1920×1080), where RGB images are in *.jpeg* format and depth images [mm] are in int16 *.png* format. The corresponding raw force array is provided in *.bin* format.

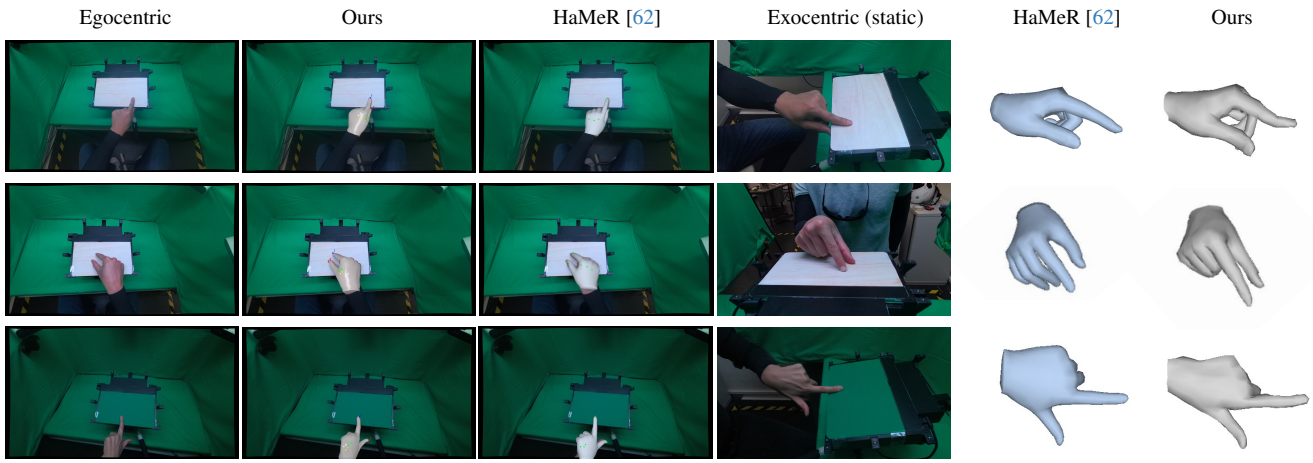


Figure 20. **Comparison of the estimated hand mesh from HaMeR [62] and our annotation method in both egocentric and exocentric views.** While the projected hand mesh from HaMeR appears visually plausible from an egocentric perspective, observable differences in hand articulation and mesh deformations become apparent from the exocentric viewpoint of the static cameras.

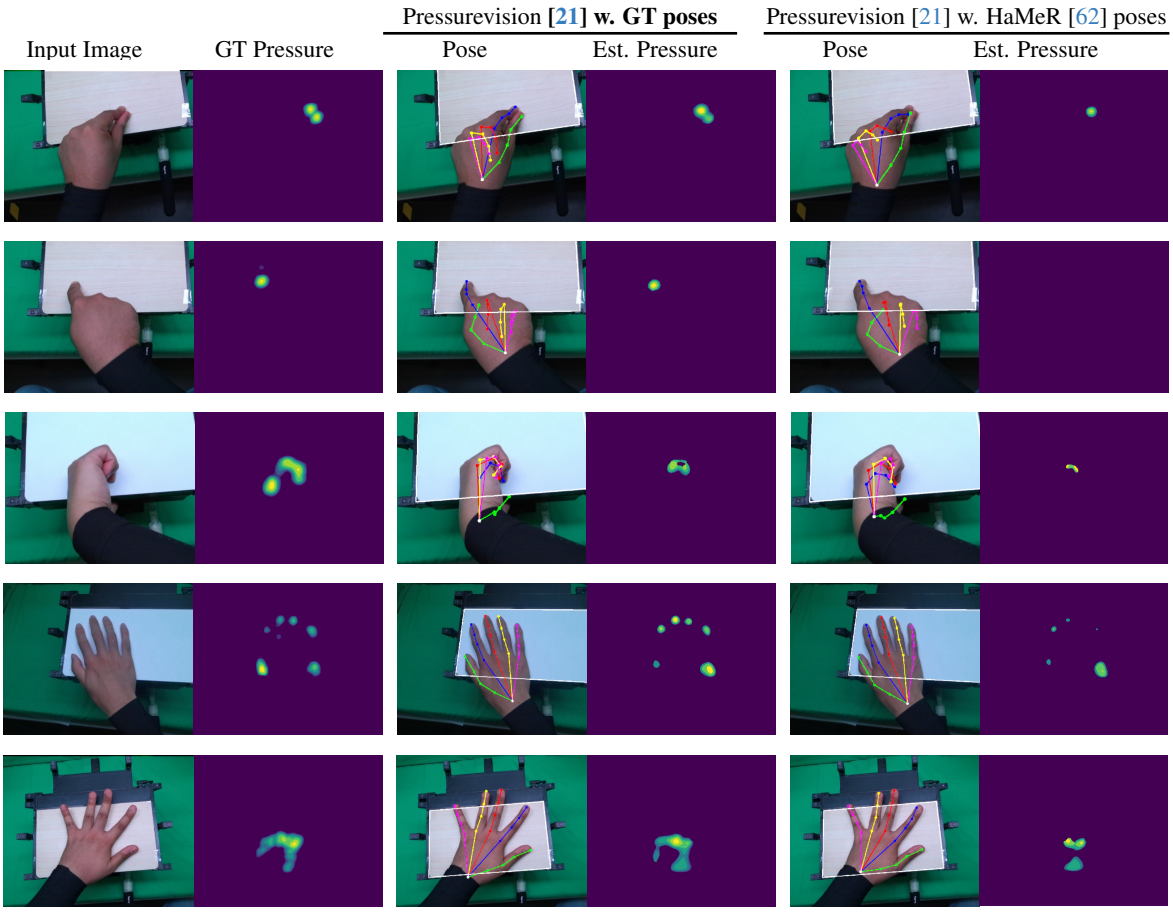


Figure 21. **Qualitative results of the image-projected baselines on egocentric views, incorporating additional hand pose inputs using our annotations and predictions from HaMeR [73].** We also reproject the area of the touchpad (indicated by white lines) to verify the egocentric camera pose.

Figure 22. **Qualitative comparison of reprojected nail tips** from our annotation method (**center**) and triangulation of HaMeR [62] predictions (**right**). The **left** column displays the reprojection of triangulated manually annotated visible tips.

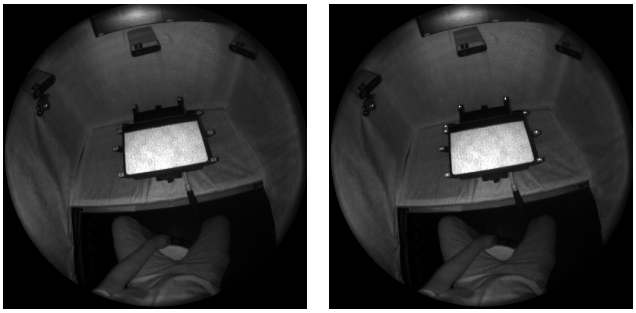


Figure 23. **Marker visibility** in infrared frame of head-mounted egocentric camera

	Gesture	Left Hand	Right Hand	Number of Repetitions
i.	calibration routine	✓	✓	-
ii.	draw word	✓	✓	3
iii.	grasp edge curled thumb-down	✓	✓	5
iv.	grasp edge curled thumb-up	✓	✓	5
v.	grasp edge uncurled thumb-down	✓	✓	5
vi.	index press high force	✓	✓	5
vii.	index press low force	✓	✓	5
viii.	index press no-contact	✓	✓	5
ix.	index press pull	✓	✓	5
x.	index press push	✓	✓	5
xi.	index press rotate left	✓	✓	5
xii.	index press rotate right	✓	✓	5
xiii.	pinch thumb-down high force	✓	✓	5
xiv.	pinch thumb-down low force	✓	✓	5
xv.	pinch thumb-down no-contact	✓	✓	5
xvi.	pinch zoom	✓	✓	5
xvii.	press cupped onebyone high force	✓	✓	3
xviii.	press cupped onebyone low force	✓	✓	3
xix.	press fingers high force	✓	✓	5
xx.	press fingers low force	✓	✓	5
xxi.	press fingers no-contact	✓	✓	5
xxii.	press flat onebyone high force	✓	✓	3
xxiii.	press flat onebyone low force	✓	✓	3
xxiv.	press palm high force	✓	✓	5
xxv.	press palm low force	✓	✓	5
xxvi.	press palm no-contact	✓	✓	5
xxvii.	press palm-and-fingers high force	✓	✓	5
xxviii.	press palm-and-fingers low force	✓	✓	5
xxix.	press palm-and-fingers no-contact	✓	✓	5
xxx.	pull towards	✓	✓	5
xxxi.	push away	✓	✓	5
xxxii.	touch iPad	✓	✓	3

Table 8. **List of gestures** performed by a participant during the data collection.

The poses of the egocentric camera for each timestamp are stored in a *.json* file for each sequence. The camera parameters and poses of all static cameras are provided in a separate *.json* meta-configuration file.

Additionally, the meta-configuration file includes basic information about the participant (gender, height, and age), handedness used during the task, and lighting conditions (camera exposure settings and the state of overhead light tubes).

We varied Kinect camera exposure (2.5 ms vs. 10 ms) and overhead lighting across three conditions: dark (2 tubes active, 2.5 ms), medium (2 tubes, 10 ms), and bright (4 tubes, 10 ms). To minimize reliance on shadows, diffuse light sources were used.

Approximately 89% of timestamps in the dataset include annotations. For each annotated timestamp, we provide a *.pkl* file containing hand pose as MANO parameters (θ, β), global translation t, vertex displacement $\boldsymbol{D}_{\text{vert}}$ with corresponding normals $\vec{\boldsymbol{n}}$, and a UV pressure map with a resolution of 224×224.

10.3. Dataset Comparisons

Table 9 provides a comprehensive comparison of our proposed dataset. Among existing public datasets focusing on contact or hand-object pose estimation, EgoPressure is the first dataset to combine egocentric video data of hand-surface interactions with ground-truth contact and pressure information, as well as high-fidelity hand poses and meshes.

Dataset	frames	participants	hand pose	hand mesh	markerless	real	egocentric	multiview	RGB	depth	contact	pressure	
												surface	hand
EgoPressure (ours)	4.3M	21	✓	✓	✓	✓	✓	✓	✓	✓	Pressure sensor	✓	✓
ContactLabelDB [22]	2.9M	51	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
PressureVisionDB [21]	3.0M	36	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
ContactPose [3]	3.0M	50	✓	✓	✓	✓	✗	✓	✓	✓	Thermal imprint	✗	✗
GRAB [77]	1.6M	10	✓	✓	✗	✓	✗	✗	✗	✗	Inferred from Pose	✗	✗
ARCTIC [17]	2.1M	10	✓	✓	✗	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
H2O [44]	571k	4	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
OakInk [84]	230k	12	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
OakInk-2 [86]	4.01M	9	✓	✓	✓	✓	✓	✓	✓	✗	Inferred from Pose	✗	✗
DexYCB [5]	582k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
HO-3D [27]	103k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
TACO [52]	5.2M	14	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
Affordpose [39]	26.7k	-	✓	✓	✓	✗	✗	✓	✗	✗	Inferred from Pose	✗	✗
AssemblyHands [61]	3.03M	34	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
ContactArt [88]	332k	-	✓	✓	✓	✗	✗	✓	✓	✓	Simulated Pose	✗	✗
HOI4D [51]	2.4M	9	✓	✓	✓	✓	✓	✗	✓	✓	Inferred from Pose	✗	✗
YCBAfford [11]	133k	-	✓	✓	✓	✗	✗	✗	✗	✗	Simulated Pose	✗	✗
ObMan [31]	154k	-	✓	✓	✓	✗	✗	✗	✓	✓	Simulated Pose	✗	✗
FPHAB [18]	100k	6	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗
HA-ViD [87]	1.5M	30	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗
Ego4d [23]	3670 hours	923	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗
EPIC-KITCHEN-100 [12]	20M	37	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗
Ego-Exo4D [24]	1422 hours	740	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗

Table 9. **Comparison between EgoPressure and extended list of hand-contact datasets.**

10.4. Details about Active IR Marker

We use active IR Marker, operating similarly to passive markers, these markers emit their own infrared light, allowing for a much smaller and more precise form factor—often appearing as tiny light dots in the filtered infrared image. This reduces the impact of lens distortion on tracking accuracy. Moreover, these markers are programmable, providing crucial control over their activation and deactivation, which is vital for synchronization within our system. We utilize the infrared led with large beam angle (see Figure 24) as active infrared marker.

An asymmetrical layout with markers can be uniquely identified from any viewpoint within the upper hemisphere above the marker arrangement. This distinctive configuration enables robust and accurate real-time tracking using filtered infrared images, where the markers appear as light dots with a radius of several pixels. The process is detailed in the pseudocode presented in Algorithm 1. The effectiveness of this layout in facilitating accurate marker identification and pose estimation is further illustrated in Figure 25, where the spatial arrangement of markers is depicted. Furthermore, this procedure can be generalized to other asymmetrical layouts.

The Perspective-n-Points (PnP) algorithm is used to compute the camera pose of the egocentric camera based on the identified markers in the infrared frame. In the experiment, the reprojection error for pose computed from well-identified markers remained below an average of 0.4 pixels. To ensure clarity and reliability in recognition, we applied a threshold value of 1 pixel to filter out frames potentially containing ambiguities in marker recognition during the recording. Additionally, for frames where tracking was lost, spherical linear interpolation (Slerp) is employed to estimate camera pose, thereby maintaining continuity and accuracy in the tracking data.

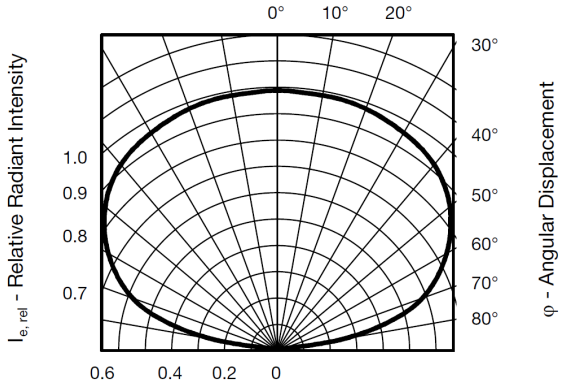


Figure 24. **Relative Radiant Intensity vs. Angular Displacement.** The marker enable a good visible radiant intensity of beam angle till 150 degree, which ensures good visibility in egocentric infrared camera.

10.5. Details about Devices' Synchronization in Dataset Acquisition

The Sensel Morph operates with zero buffer and maintains a stable 8 ms delay at 120 fps, whereas the Azure Kinect cameras function at 30 fps, capturing high-resolution RGB images and a depth map. Due to the high recording performance of the Azure Kinect, frames are initially stored in the device's cache, making it impractical to rely on the OS timestamp at the frame's arrival on the host computer for synchronization with Sensel Morph pressure data.

All cameras can be externally synchronized via a 30 Hz triggering signal from the Raspberry Pi CM4, ensuring simultaneous frame capture. However, an initial frame loss (1–3 frames) occurs at the start of recording due to device-specific issues. Since the absolute value of device ticks has no inherent meaning, it is unclear how many frames were lost before the first received frame. Relying on device tick

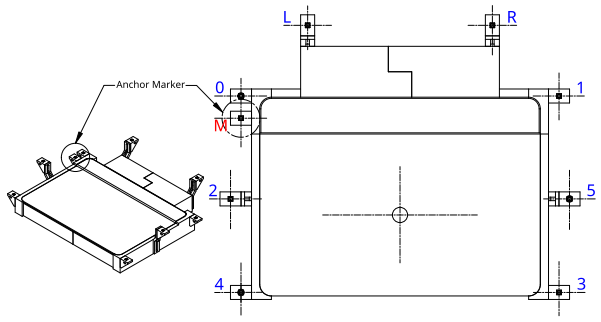


Figure 25. **Layout of the Active Markers.** The indices of the markers are aligned with the pseudocode provided in Algorithm 1. Starting from the asymmetrical anchor marker **M**, all markers can be identified by computing their relative distances and considering their spatial relationships.

Algorithm 1 Identify Marker

procedure IDENTIFYMARKERS(filtered IR image)
  Extract marker coordinates (u, v) from the filtered IR image
  Compute all pairwise distances among markers
  Identify the pair with the smallest distance, initially labeled as 0 and M
  Compute the vector from M to 0
  Count the number of markers on each side of the vector line $M - 0$
  if more markers lie on the right of the vector **then**
    Confirm start point as M, endpoint as 0
  else
    Swap, set start point as 0 and endpoint as M
  end if
  Identify 2 and 4 as markers aligned with $M - 0$, on the same side relative to M
  Check distances from M to 2 and 4 to determine which is closer
  Identify L as the marker closest to the line extending through $(0, M, 2, 4)$ and on the same side as 0
  Compute the centroid of all markers
  Draw a line from 0 through the centroid
  Identify 3 as the marker isolated on its side of the centroid line
  Identify 5 as the closest marker to the line $(0-\text{centroid})$ not already labeled
  Determine 1 and R by their proximity to line $(2 - 5)$, with 1 being closer
end procedure

differences for synchronization could therefore introduce a misalignment of 1–3 frames between cameras.

To address this, the programmable features of active infrared markers and the precise global OS timestamp synchronization (within 1 ms) between the two host computers and the Raspberry Pi CM4, facilitated by the Precision Time Protocol (PTP), are utilized. The Raspberry Pi CM4, equipped with basic electrical components (see Figure 23) at the start of the next exposure cycle, providing a reliable synchronization point that compensates for the initial missing frames. The exact global OS timestamp of the marker activation is clearly recorded (see Figure 27).

By calculating the real OS timestamp for all frames based on the offset from device ticks, starting from the frame where the marker first appears, precise synchronization is achieved. This approach effectively aligns RGBD images and pressure data, optimizing data integration across the multi-modal sensor system. Moreover, this synchronization mechanism using an external active optical identifier is efficient and economical, making it generalizable to other multi-sensor systems, such as motion capture systems with external head-mounted cameras, that rely on different OS timestamp sources.

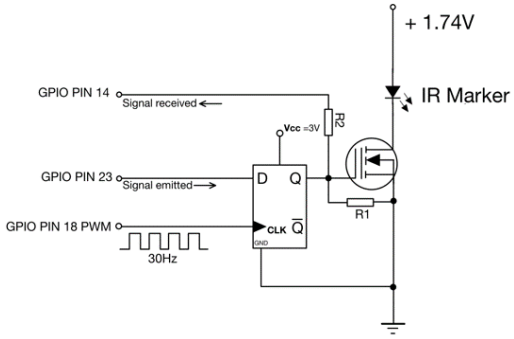


Figure 26. **Basic Electrical Elements Implementation.** We use a D-type flip-flop and N-channel MOSFET to ensure the IR marker will be activate by the next beginning of exposure after receiving signal from PIN 23. And PIN 14 will monitor the activation to obtain its timestamp.

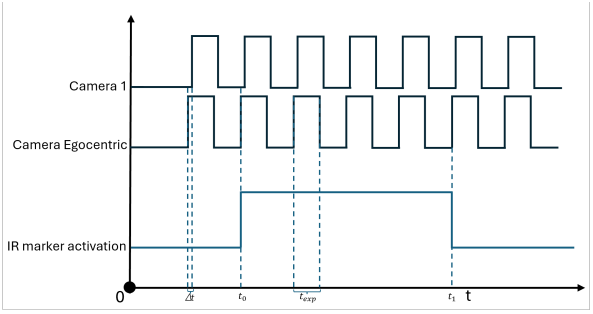


Figure 27. **Synchronization Diagram** We set head-mounted egocentric camera to align with 30 Hz triggering signal emitted by the Raspberry Pi CM4, this signal will also go to PIN 18 as clock frequency of D-type Flip-flop Fig. 26). Then exposure t_{exp} of all cameras is same. The other static cameras 1 to 7 will have a delay $\Delta\, t$ to triggering signal to avoid interference of infrared light. The marker will be activate at t_0 (around 300 milliseconds after start recording), which we know its global OS timestamp, then it will be visible to all camera at next exposure cycle. As verification, we deactivate marker by the very end of recording at the timestamp t_1, then the marker will be invisible for all cameras in the next frame capture. The good synchronization will have equal frame number between t_0 and t_1 for all cameras.

11. Limitations

Although EgoPressure serves as a foundational study for understanding pressure from an egocentric view, several challenges remain unresolved. These challenges are categorized into three main areas.

First, measuring pressure while interacting with general objects presents a challenge. Our current data capture is confined to sensing pressure on flat surfaces. While we are optimistic that future research will expand to include a wider variety of objects, sensing pressure on arbitrary surfaces poses significant challenges, as it would require extensive instrumentation of the user's hands, hindering natural interaction and introducing visible artifacts in the captured data. Instrumenting objects for pressure sensing remains an ongoing research area, with recent advancements primarily in basic contact detection [3]. However, we anticipate that our annotation method will extend naturally to more complex objects and interactions as these challenges are addressed. PressureVision++ [22] explores weak labels to infer pressure on more complex objects. However, it only considers fingertip interactions and its evaluation of pressure regression remains limited to flat surfaces due to the challenges of acquiring precise pressure. We present a qualitative evaluation of PressureFormer on a wider variety of objects in Figure 16.

Second, the current dataset was only captured in an indoor setting. Our data capture setup is optimized for acquiring high-fidelity annotations of hand-surface interactions. To increase the diversity of background environments to improve generalization to real-world settings, we have added green overlays to the background of our data capture rig and to the pressure pad. This allows for background replacement (see Figure 28) and has been successfully demonstrated to enhance commercial in-the-wild hand tracking [30, 89].

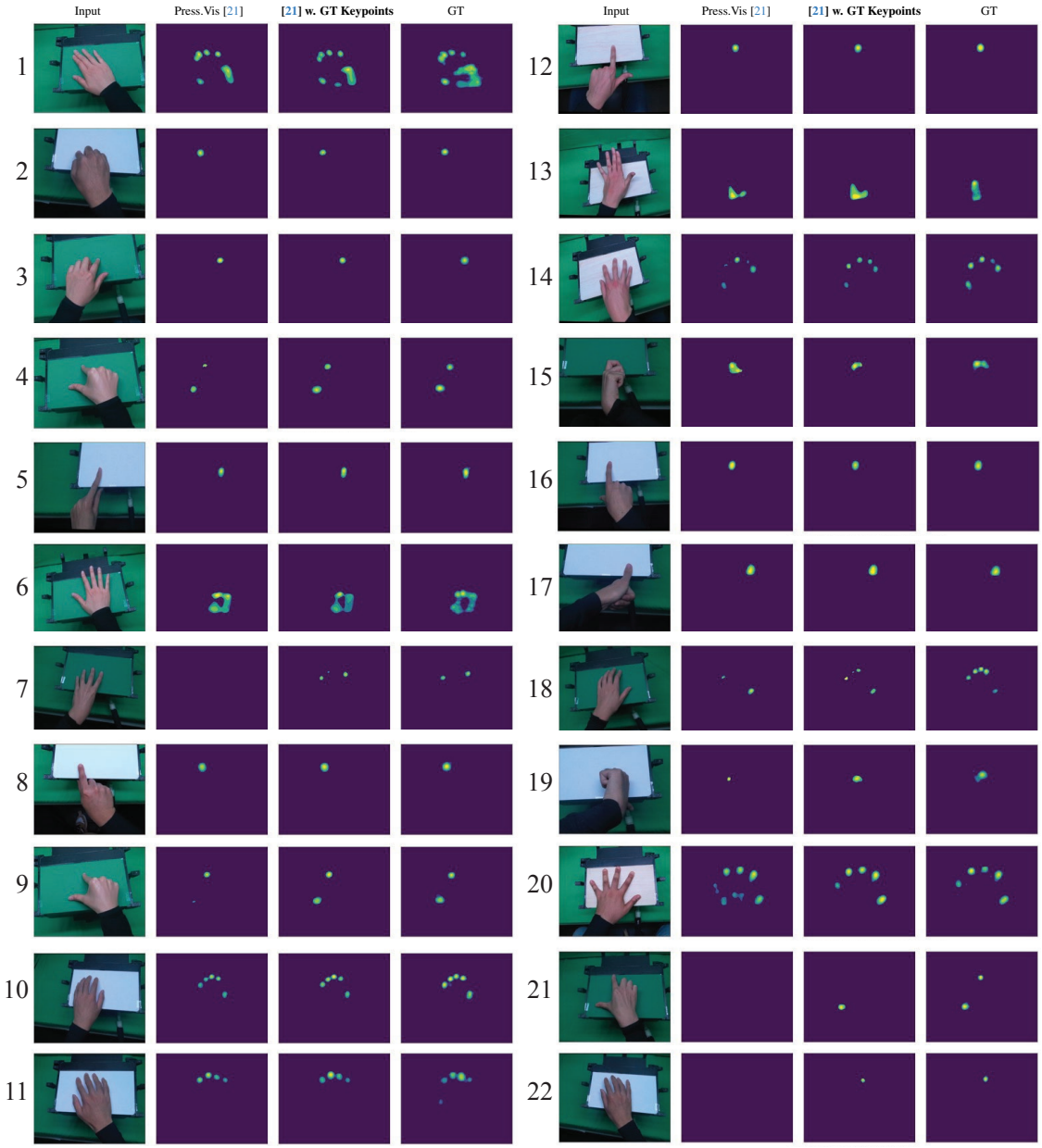
Finally, the current setup only considers single-hand interactions. Incorporating scenarios involving the use of both hands would be a natural extension of our work.

Further addressing these challenges in future research would improve pressure estimation in real-world scenarios and broaden its applicability.

Figure 28. **Examples of background augmentation using hand masks and a touchpad.**

12. Ethical Considerations

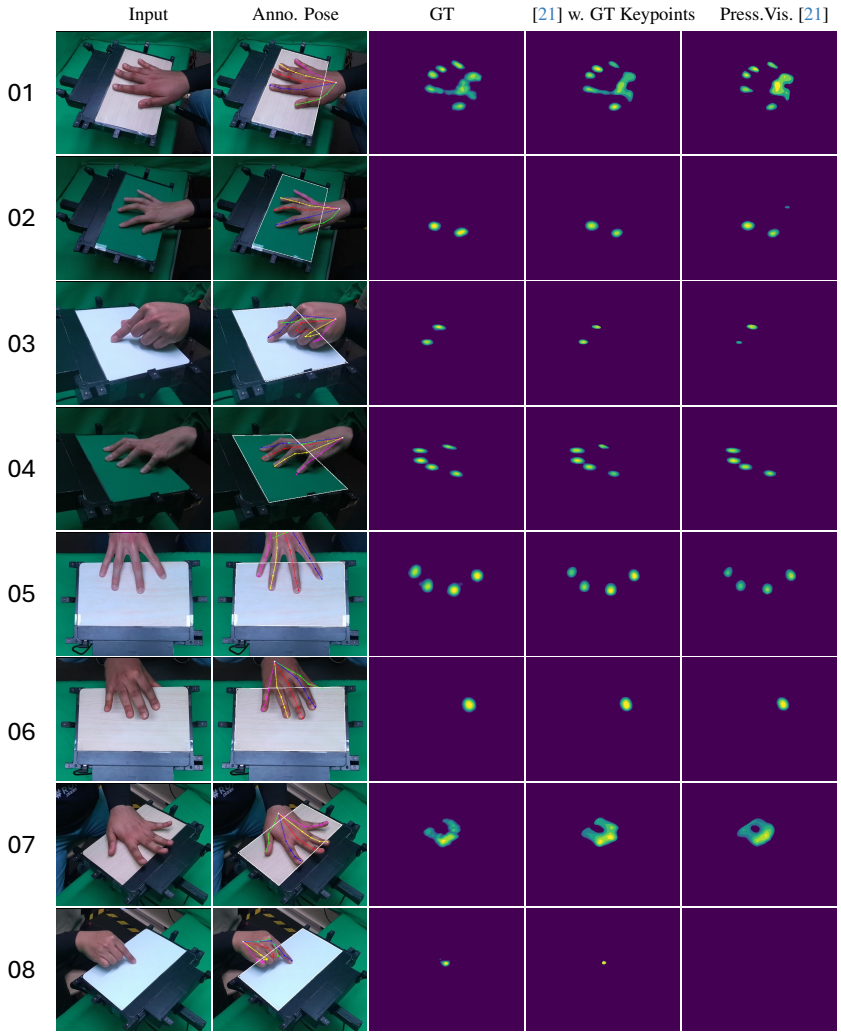
The recording and use of human activity data involve important ethical considerations. The EgoPressure project has received approval from ETH Zürich Ethics Commission as proposal EK 2023-N-228. This approval includes both the data collection and the public release of the dataset. All participants provided explicit written consent for recording their sessions, creating the dataset, and releasing it (see accompanying consent form). All demographic information (such as sex, age, weight, and height) along with the sensor and video data are pseudonymized, assigning a numeric code to each participant. Personal data (sex, age, weight, and height) is stored separately from the sensor and video data, and is accessible only to the primary researchers involved in the study. We have not captured or stored any images of the participant's face.



		1	2	3	4	5	6	7	8	9	10	11
MAE ↓	**Press.Vis. [21]**	129.6	8.2	14.3	25.3	6.7	61.9	9.3	8.3	23.9	39.1	37.1
	[21] w. GT poses	116.4	5.3	11.8	17.3	4.8	62.2	5.1	6.2	13.5	30.6	35.6
Contact IoU ↑	**Press.Vis. [21]**	46.9	68.4	57.4	39.0	84.0	63.8	0.0	76.9	35.1	64.1	56.8
	[21] w. GT poses	55.2	79.9	63.9	67.8	86.8	65.6	58.7	82.6	73.2	72.1	65.4

		12	13	14	15	16	17	18	19	20	21	22
MAE ↓	**Press.Vis. [21]**	6.2	50.1	39.4	54.7	11.7	10.7	32.0	37.0	67.4	21.5	8.29
	[21] w. GT Keypoints	4.7	51.7	35.6	30.4	10.0	10.3	24.9	32.6	35.5	14.7	4.89
Contact IoU ↑	**Press.Vis. [21]**	86.2	38.4	54.2	24.2	79.9	76.6	17.6	1.2	54.1	0.0	0.0
	[21] w. GT Keypoints	86.3	48.4	61.7	43.8	83.1	79.6	30.3	35.2	76.3	42.2	43.6

Figure 29. **Qualitative comparison of pressure maps** inferred using PressureVisionNet [21] and our trained model with additional hand poses as input on representative cases across various gestures. The bottom table presents MAE [Pa] and Contact IoU [%] for pressure maps inferred using PressureVisionNet [1] and our trained model on selected samples shown in the Figure.



		01	02	03	04	05	06	07	08
MAE [Pa] ↓	**Press.Vis. [21]**	136.5	114.8	22.3	57.1	56.4	51.3	71.6	22.1
	[21] w. GT Keypoints	117.0	106.2	20.5	50.5	49.9	46.4	50.6	9.1
Contact IoU [%] ↑	**Press.Vis. [21]**	65.6	74.9	51.9	63.1	59.2	77.8	59.5	0.0
	[21] w. GT Keypoints	78.1	77.4	58.1	69.2	65.1	83.2	67.7	31.1
	Camera #	2	2	3	3	4	4	5	5

Figure 30. **Comparison of pressure maps** estimated by PressureVisionNet [21] and our adapted model, using separate training and validation sets, both consisting of images from camera views 2, 3, 4, and 5.

		09	10	11	12	13	14
MAE ↓	**Press.Vis. [21]**	52.8	77.4	119.5	76.5	62.6	199.6
	[21] w. GT keypoints	23.4	70.8	102.7	43.7	52.8	182.0
Contact IoU ↑	**Press.Vis. [21]**	0.0	43.6	60.1	17.4	48.2	49.3
	[21] w. GT keypoints	38.5	56.0	68.9	43.5	55.7	57.1
	Camera #	1	1	7	7	6	6

Figure 31. **Comparison of pressure maps** estimated by PressureVisionNet [21] and our adapted model, evaluated using input images from cameras 1, 6, and 7. The models are the same as in Figure 30, which are trained on images from camera views 2, 3, 4, and 5.

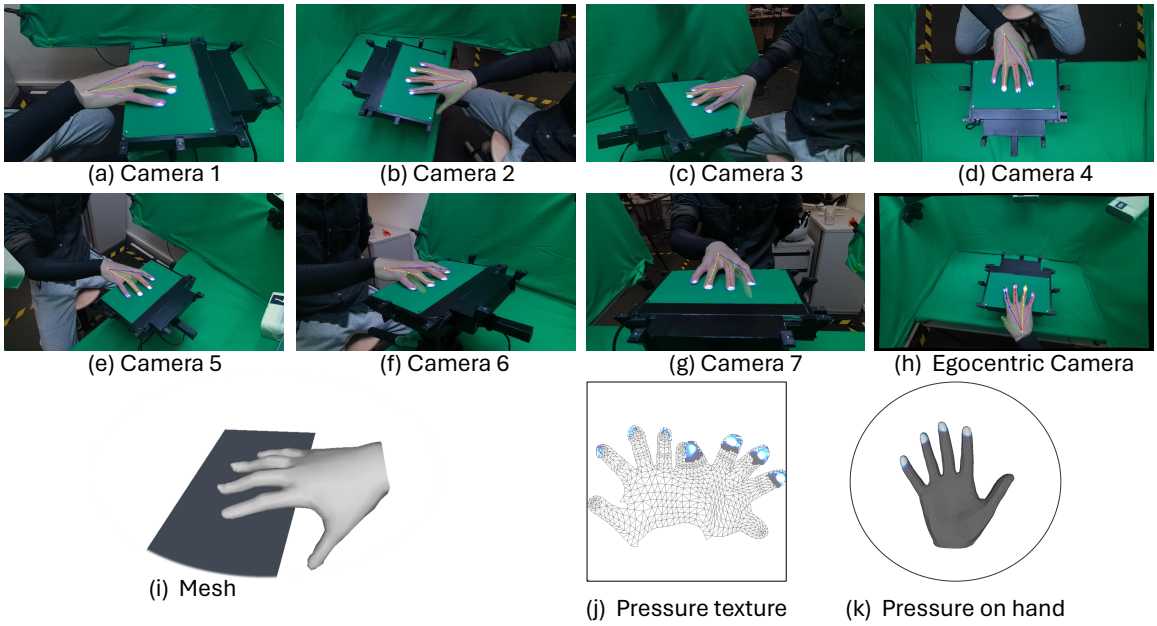


Figure 32. **Example of Annotation 1.** Right hand with gesture: grasp edge with uncurled thumb down.

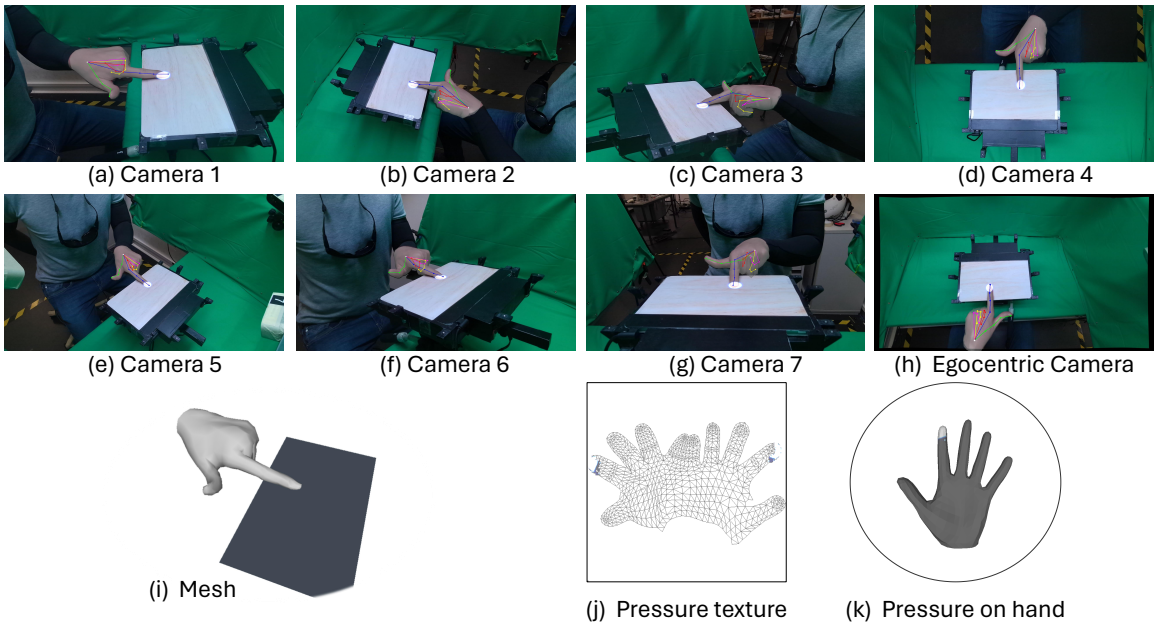


Figure 33. **Example of Annotation 2.** Left hand with gesture: index press with high force.

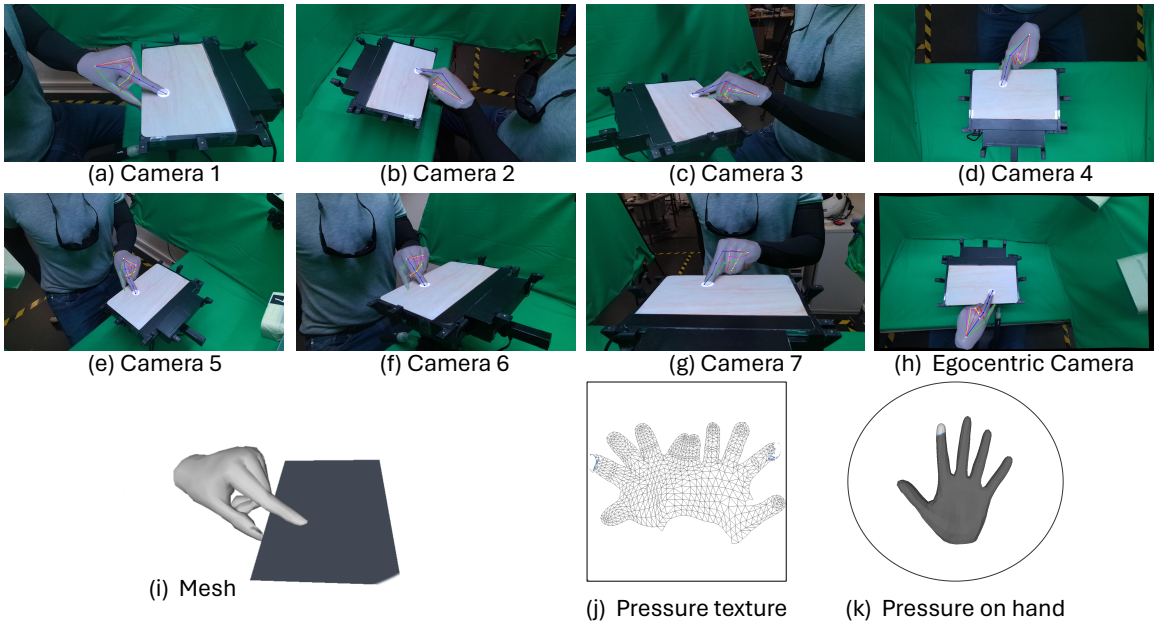


Figure 34. **Example of Annotation 3.** Left hand with gesture: pinch thumb down on the edge with high force.

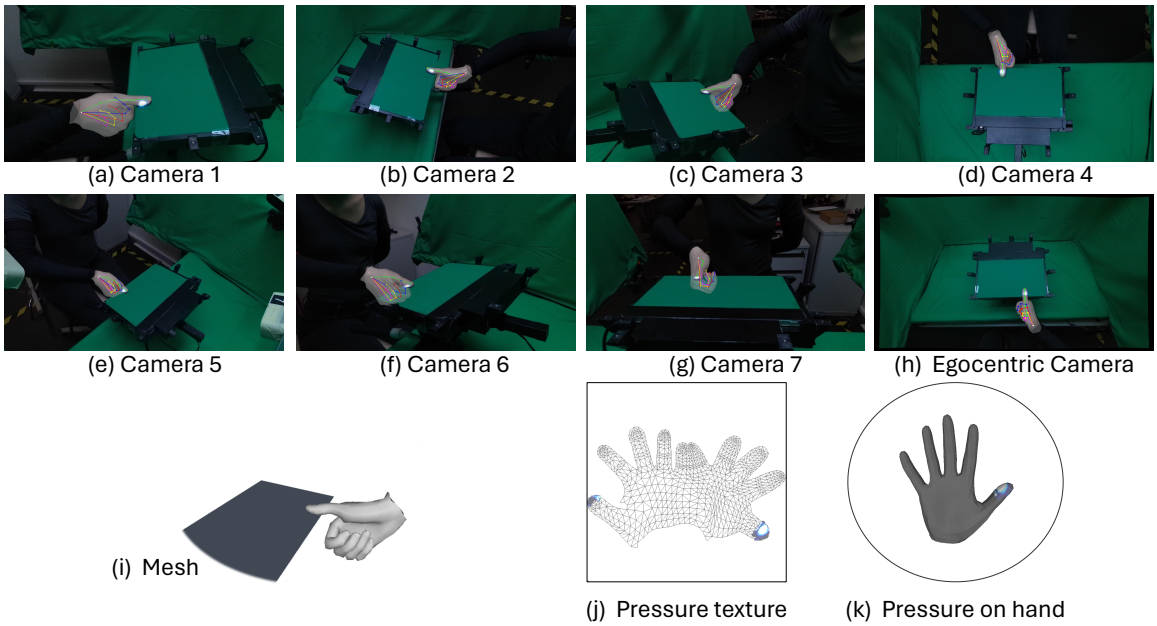


Figure 35. **Example of Annotation 4.** Right hand with gesture: grasp edge with curled thumb up.

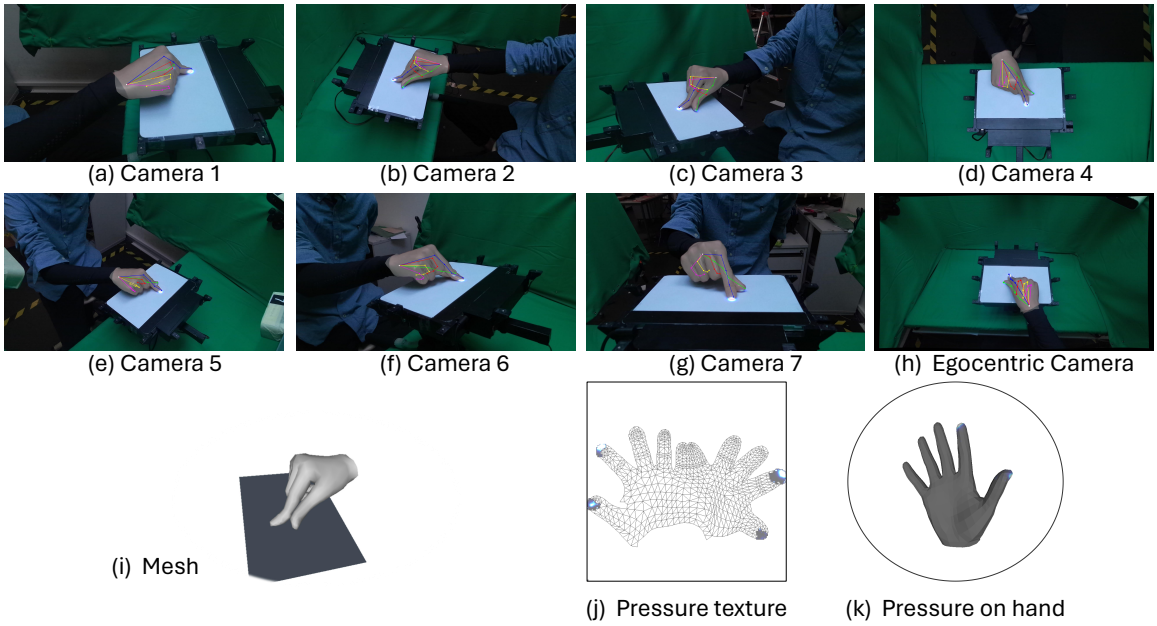


Figure 36. **Example of Annotation 5.** Right hand with gesture: pinch finger zoom in and out.

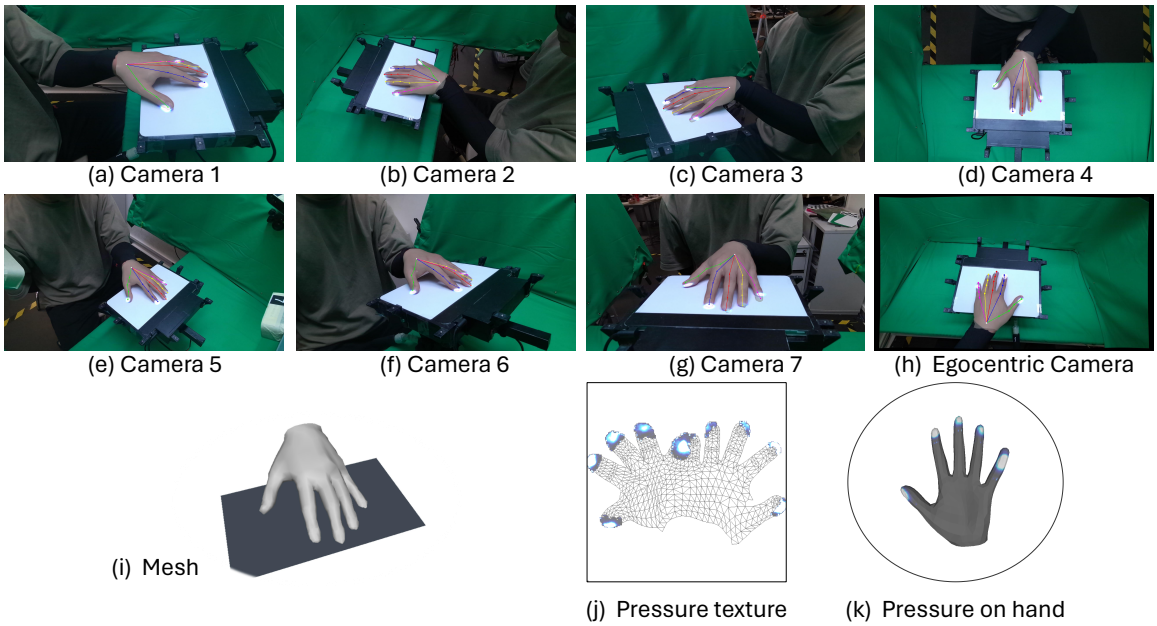


Figure 37. **Example of Annotation 6.** Left hand with gesture: pull all fingers towards the participant.

References

[1] Azure Kinect. Azure kinect dk hardware specifications, 2019. 5

[2] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin: versatile, replaceable, lasting tactile skins. In *5th Annual Conference on Robot Learning*, 2021. 2

[3] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. Contactpose: A dataset of grasps with object contact and hand pose. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 361–378. Springer, 2020. 2, 3, 8, 10

[4] Gereon H Büscher, Risto Kõiva, Carsten Schürmann, Robert Haschke, and Helge J Ritter. Flexible and stretchable fabric-based tactile sensor. *Robotics and Autonomous Systems*, 63: 244–252, 2015. 2

[5] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021. 2, 3, 8

[6] Nutan Chen, Göran Westling, Benoni B Edin, and Patrick van der Smagt. Estimating fingertip forces, torques, and local curvatures from fingernail images. *Robotica*, 38(7):1242–1262, 2020. 2

[7] Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances In Neural Information Processing Systems*, 2019. 3, 7, 8

[8] Yi Fei Cheng, Tiffany Luong, Andreas Rene Fender, Paul Streli, and Christian Holz. Comfortable user interfaces: Surfaces reduce input error, time, and exertion for tabletop and mid-air user interfaces. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022. 1, 2

[9] Sammy Christen, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges. D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20577–20586, 2022. 1, 2

[10] Jeremy A Collins, Cody Houff, Patrick Grady, and Charles C Kemp. Visual contact pressure estimation for grippers in the wild. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10947–10954. IEEE, 2023. 1

[11] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5031–5041, 2020. 2, 8

[12] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 8

[13] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999. 3

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[15] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2020. 2

[16] Neil Xu Fan and Robert Xiao. Reducing the latency of touch tracking on ad-hoc surfaces. *Proc. ACM Hum.-Comput. Interact.*, 6(ISS), 2022. 2

[17] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12954, 2023. 2, 3, 8

[18] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 409–419, 2018. 2, 8

[19] Jun Gong, Aakar Gupta, and Hrvoje Benko. Acustico: Surface tap detection and localization using wrist-based acoustic tdoa sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, page 406–419, New York, NY, USA, 2020. Association for Computing Machinery. 2

[20] Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1471–1481, 2021. 2

[21] Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. Pressurevision: Estimating hand pressure from a single rgb image. In *European Conference on Computer Vision*, pages 328–345. Springer, 2022. 1, 2, 3, 6, 7, 8, 11, 12, 13

[22] Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. Pressurevision++: Estimating fingertip pressure from diverse rgb images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8698–8708, 2024. 1, 2, 3, 6, 7, 8, 10

[23] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger,

Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1, 8

[24] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023. 1, 2, 8

[25] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. Accurate and low-latency sensing of touch contact on any surface with finger-worn imu sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, page 1059–1070, New York, NY, USA, 2019. Association for Computing Machinery. 2

[26] Sean Gustafson, Christian Holz, and Patrick Baudisch. Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, page 283–292, New York, NY, USA, 2011. Association for Computing Machinery. 2

[27] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020. 2, 3, 8

[28] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint Transformer: Solving Joint Identification in Challenging Hands and Object Interactions for Accurate 3D Pose Estimation . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11080–11090, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 2

[29] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)*, 39(4):87–1, 2020. 2

[30] Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, et al. Umetrack: Unified multi-view end-to-end hand tracking for vr. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 10

[31] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019. 2, 8

[32] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 3

[33] Steven Henderson and Steven Feiner. Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010. 1

[34] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1

[35] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019. 1

[36] Wonjun Hwang and Soo-Chul Lim. Inferring interaction force from visual information without using physical force sensors. *Sensors*, 17(11):2455, 2017. 2

[37] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019. 1

[38] Sensel Inc. Sensel morph., 2024. 2, 4

[39] Juntao Jian, Xiuping Liu, Manyi Li, Ruizhen Hu, and Jian Liu. Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14713–14724, 2023. 8

[40] Tero Karras. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics*, pages 33–37. Eurographics Association, 2012. 4, 3

[41] Korrawe Karunratanakul, Sergey Prokudin, Otmar Hilliges, and Siyu Tang. HARP: Personalized Hand Reconstruction from a Monocular RGB Video. 2023. 3

[42] Hong-Ki Kim, Seunggun Lee, and Kwang-Seok Yun. Capacitive tactile sensor array for touch screen application. *Sensors and Actuators A: Physical*, 165(1):2–7, 2011. 2

[43] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 3, 2

[44] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10138–10148, 2021. 2, 3, 8

[45] Mike Lambeta, Tingfan Wu, Ali Sengul, Victoria Rose Most, Nolan Black, Kevin Sawyer, Romeo Mercado, Haozhi Qi, Alexander Sohn, Byron Taylor, et al. Digitizing touch with an artificial multimodal fingertip. *arXiv preprint arXiv:2411.02479*, 2024. 1

[46] Minkyung Lee, Woontack Woo, et al. Arkb: 3d vision-based augmented reality keyboard. In *ICAT*, 2003. 2

[47] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic. Estimating 3d motion and forces of person-object interactions from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8640–8649, 2019. 2

[48] Chen Liang, Xutong Wang, Zisu Li, Chi Hsia, Mingming Fan, Chun Yu, and Yuanchun Shi. Shadowtouch: Enabling free-form touch-based hand-to-surface interaction with wrist-mounted illuminant by shadow projection. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14, 2023. 2

[49] PPS UK Limited. Tactileglove - hand pressure and force measurement., 2023. 2

[50] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[51] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 2, 8

[52] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. *arXiv preprint arXiv:2401.08399*, 2024. 3, 8

[53] Yiyue Luo, Yunzhu Li, Pratyusha Sharma, Wan Shou, Kui Wu, Michael Foshey, Beichen Li, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. Learning human–environment interactions using conformal tactile textiles. *Nature Electronics*, 4(3):193–201, 2021. 1, 2

[54] Yiyue Luo, Chao Liu, Young Joong Lee, Joseph DelPreto, Kui Wu, Michael Foshey, Daniela Rus, Tomás Palacios, Yunzhu Li, Antonio Torralba, et al. Adaptive tactile interaction transfer via digitally embroidered smart gloves. *Nature communications*, 15(1):868, 2024. 1

[55] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 6169–6176. IEEE, 2021. 1, 2

[56] Stephen A Mascaro and H Harry Asada. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *IEEE Transactions on Robotics and Automation*, 20(1):26–35, 2004. 2

[57] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. Tapid: Rapid touch interaction in virtual reality using wearable sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 519–528. IEEE, 2021. 1, 2

[58] Vimal Mollyn and Chris Harrison. Egotouch: On-body touch input using ar/vr headset cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–11, 2024. 2

[59] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020. 2

[60] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1163, 2017. 2

[61] Takehiko Ohkawa, Kun He, Fadime Sener, Tomas Hodan, Luan Tran, and Cem Keskin. Assemblyhands: Towards egocentric activity understanding via 3d hand pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12999–13008, 2023. 8

[62] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024. 2, 3, 6, 7, 8, 4, 5

[63] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. Hand-object contact force estimation from markerless visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2883–2896, 2017. 2

[64] Chandradeep Pokhariya, Ishaan Nikhil Shah, Angela Xing, Zekun Li, Kefan Chen, Avinash Sharma, and Srinath Sridhar. Manus: Markerless grasp capture using articulated 3d gaussians. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2197–2208, 2024. 2

[65] Philip Quinn, Wenxin Feng, and Shumin Zhai. Deep touch: Sensing press gestures from touch image sequences. *Artificial Intelligence for Human Computer Interaction: A Modern Approach*, pages 169–192, 2021. 2

[66] Mark Richardson, Matt Durasoff, and Robert Wang. Decoding surface touch typing from hand-tracking. In *Proceedings of the 33rd annual ACM symposium on user interface software and technology*, pages 686–696, 2020. 2

[67] Mark Richardson, Fadi Botros, Yangyang Shi, Pinhao Guo, Bradford J Snow, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. Stegotype: Surface typing from egocentric cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14, 2024. 2

[68] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017. 2, 3, 7

[69] Pressure Mapping Sensors. Tekscan., 2024. 2

[70] Yilei Shi, Haimo Zhang, Jiashuo Cao, and Suranga Nanayakkara. Versatouch: A versatile plug-and-play system that enables touch interactions on everyday passive surfaces. In *Proceedings of the Augmented Humans International Conference*, New York, NY, USA, 2020. Association for Computing Machinery. 2

[71] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. Ready, steady, touch! sensing physical contact with a finger-mounted imu. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(2), 2020. 2

[72] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007. 3

[73] Paul Streli, Jiaxi Jiang, Andreas Fender, Manuel Meier, Hugo Romat, and Christian Holz. Taptype: Ten-finger text entry on everyday surfaces via bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2022. Association for Computing Machinery. 2

[74] Paul Streli, Jiaxi Jiang, Juliete Rossie, and Christian Holz. Structured light speckle: Joint ego-centric depth estimation and low-latency contact detection via remote vibrometry. In

Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, pages 1–12, 2023. 2

[75] Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. Touchinsight: Uncertainty-aware rapid touch and text input for mixed reality from egocentric vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–16, 2024. 1, 2

[76] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, 2019. 2

[77] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020. 2, 3, 8

[78] Ryo Takahashi, Masaaki Fukumoto, Changyo Han, Takuya Sasatani, Yoshiaki Narusue, and Yoshihiro Kawahara. Telemetring: A batteryless and wireless ring-shaped keyboard using passive inductive telemetry. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, page 1161–1168, New York, NY, USA, 2020. Association for Computing Machinery. 2

[79] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 118 (2):172–193, 2016. 4, 3

[80] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 5

[81] Andrew D Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 83–92, 2005. 1, 2

[82] Andrew D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, page 69–72, New York, NY, USA, 2010. Association for Computing Machinery. 2

[83] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. Mrtouch: Adding touch input to head-mounted mixed reality. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1653–1660, 2018. 2

[84] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20953–20962, 2022. 2, 3, 8

[85] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4866–4874, 2017. 2

[86] Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. *arXiv preprint arXiv:2403.19417*, 2024. 2, 3, 8

[87] Hao Zheng, Regina Lee, and Yuqian Lu. Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding. *Advances in Neural Information Processing Systems*, 36, 2024. 8

[88] Zehao Zhu, Jiashun Wang, Yuzhe Qin, Deqing Sun, Varun Jampani, and Xiaolong Wang. Contactart: Learning 3d interaction priors for category-level articulated object and hand poses estimation. *arXiv preprint arXiv:2305.01618*, 2023. 2, 8

[89] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2019. 2, 10

[90] Lara Zlokapa, Yiyue Luo, Jie Xu, Michael Foshey, Kui Wu, Pulkit Agrawal, and Wojciech Matusik. An integrated design pipeline for tactile sensing robotic manipulators. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3136–3142. IEEE, 2022. 1