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

Supplementary Material

S1. Details about Benchmark Evaluation

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

S1.1. Details for image-projected pressure baselines

S1.1.1. Baseline model with Additional Keypoint depth maps

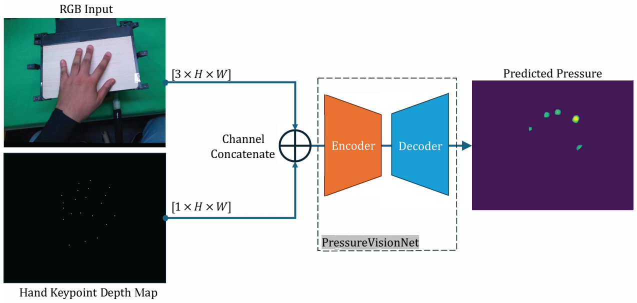


Figure S1. **Overview of the image-projected pressure baseline with additional hand pose input.** The baseline receives an RGB image and a keypoint depth map as inputs to an encoder-decoder segmentation network for pressure estimation.

The previous method [8] 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) [15, 16] as the encoder, which takes both RGB and 3D hand pose inputs, and a feature pyramid network [17, 22] 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.

S1.1.2. Evaluation Metrics

For evaluation, we adopt the four metrics proposed in PressureVision [8]: 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.

S1.2. Additional Qualitative Results

More qualitative results for the baselines are provided in Figure S18. More qualitative examples for the annotations are shown in Figures S21, S22, S23, S24, S25 and S26.

We also present qualitative results from the third-person view camera experiments (refer to Table 2 in the main paper). Figure S19 and S20 include visual comparisons between our model, which uses RGB and a hand keypoint depth map, and PressureVisionNet [8] which uses only RGB input. Figure S19 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 S19, 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 S19 and Figure S20, 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 S19 and *09, 11, 13* in Figure S20).

S1.3. 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 [8, 9], it offers additional insights about the specific hand regions applying pressure.

$$\mathcal{L}_{\mathcal{R}}(\boldsymbol{\Theta}) = \sum_{i=0}^{C} [\lambda_M \underbrace{(1 - \mathrm{IoU}(\mathcal{R}_M^i(\boldsymbol{\Theta}), \boldsymbol{M}_{\text{gt}}^i))}_{\textbf{Mask IoU Loss } \mathcal{L}_M(\boldsymbol{\Theta})} + \lambda_A \underbrace{\mathrm{MSE}(\mathcal{R}_F^i(\boldsymbol{\Theta}, \boldsymbol{\mathcal{T}}), \boldsymbol{I}_{\text{gt}}^i)}_{\textbf{Appearance Loss } \mathcal{L}_A(\boldsymbol{\Theta})} + \lambda_D \underbrace{(1 - \frac{|\min(\mathcal{R}_D^i(\boldsymbol{\Theta}), \boldsymbol{D}_{\text{gt}}^i)|_1}{|\max(\mathcal{R}_D^i(\boldsymbol{\Theta}), \boldsymbol{D}_{\text{gt}}^i)|_1})}_{\textbf{Depth Volumetric IoU Loss } \mathcal{L}_D(\boldsymbol{\Theta})}]. \tag{S1}$$

This capability is beneficial for scenarios involving 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 [8] 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 $w_2 = 0.05$.

S2. Details and Evaluation of Annotation Method

S2.1. Optimization Objectives

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

S2.1.1. 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) [20] 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. S1

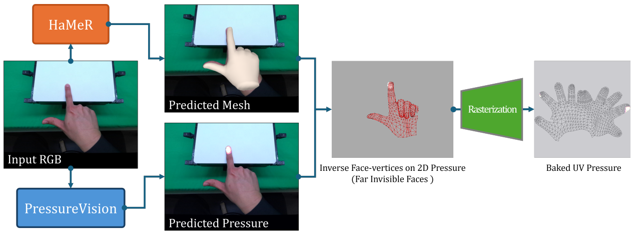


Figure S2. **Pipeline for projecting the image-based pressure map (from PressureVision) onto the UV map:** Starting with the predicted hand mesh and 2D pressure map, the normals and z-axis are inverted to identify occluded (invisible) faces of the mesh. The pressure is then mapped onto the UV space using rasterization.

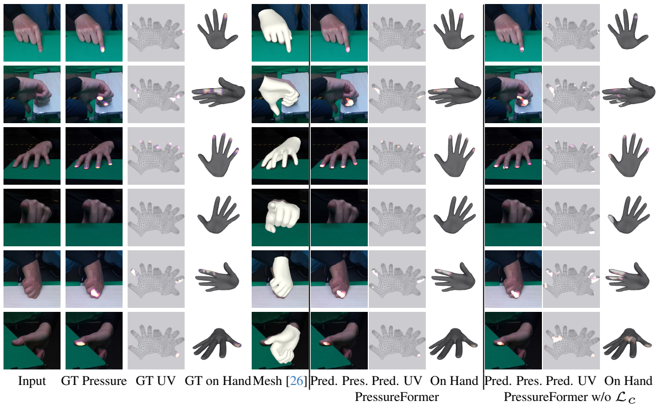


Figure S3. **Qualitative examples demonstrating the impact of coarse UV loss supervision $\mathcal{L}_c$.** The coarse UV loss supervision $\mathcal{L}_c$ prevents the prediction of pressure in areas of the UV map that are not rendered on the image plane (see Section 5.2). These regions typically correspond to faces oriented toward the camera, where pressure and contact are not physically possible.

Depth Volumetric IoU $\mathcal{L}_D(\boldsymbol{\Theta})$ [8] is defined in the third term of Equation S1. We apply it to the ground truth and rendered depth. In Table S1, 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

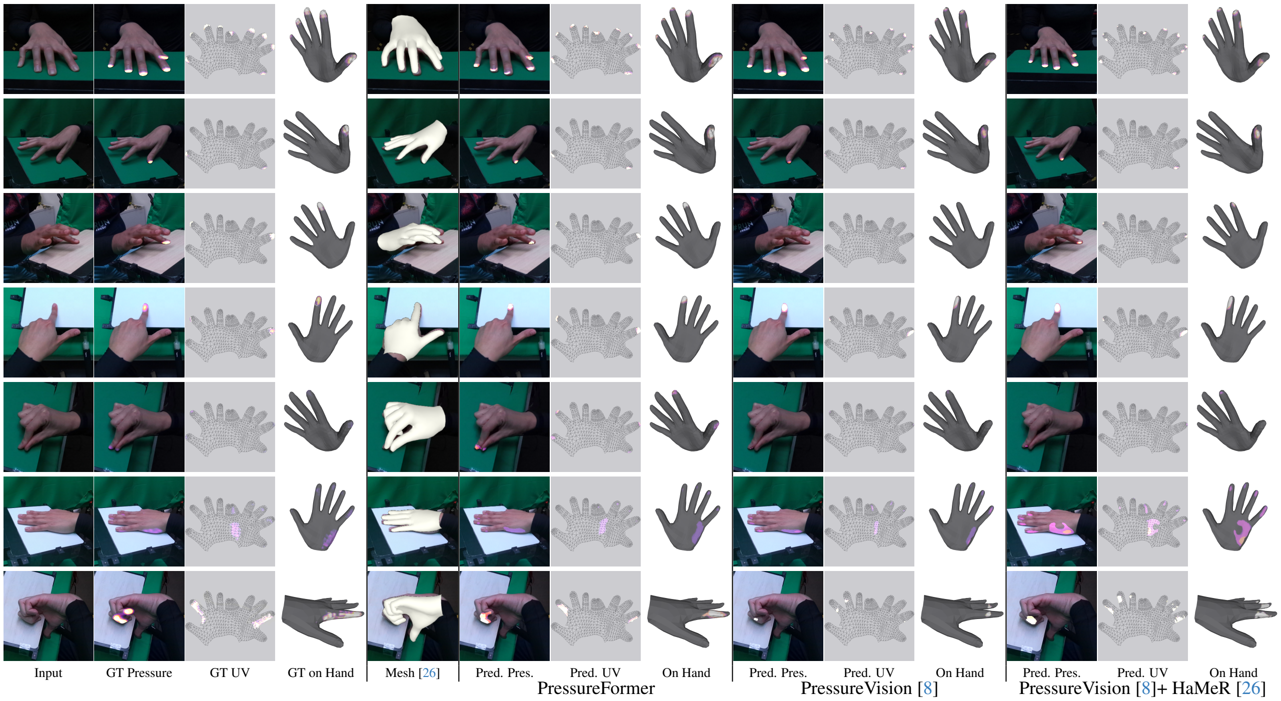


Figure S4. **Qualitative comparison of UV Pressure.** We compare our PressureFormer model against the original PressureVision [8] and its extended version with additional hand keypoint inputs. For both PressureVision-based approaches, the UV pressure is obtained by baking the image-based pressure predictions onto the UV map of the hand mesh, using the hand mesh estimates provided by HaMeR [26].

two consecutive annotation stages, Pose Optimization and Shape Refinement.

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 S1. **Losses by Stages.** We validate the quality of hand poses using two metrics, Depth Volumetric IoU Loss $\mathcal{L}_D$ (Eq. S1) and Mask IoU Loss $\mathcal{L}_M$ (Eq. S1), computed on 386,231 ×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.

S2.1.2. Geometry Objective

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

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L}_{\text{insec}} + \mathcal{L}_{\text{arap}} + \mathcal{L}_{\vec{\mathbf{n}}} + \mathcal{L}_{\text{lap}} + \mathcal{L}_{\text{offset}} \qquad \text{(S2)}$$

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 [19, 29].

The term $\mathcal{L}_{\text{arap}}$, as-rigid-as-possible loss, as introduced in [27], 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

$$\mathcal{L}_{\text{arap}} = \frac{1}{|E|} \sum_{v^* \in \boldsymbol{\Theta}^*} \sum_{\boldsymbol{e}^* \in E(v^*, u^*)} \left| \|\boldsymbol{e}^*\| - \|\boldsymbol{e}^p\| \right|, \qquad \text{(S3)}$$

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 $\mathcal{L}_{\text{lap}}$ [5], 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 $\|\boldsymbol{D}_{\text{vert}}\|^2$.

S2.1.3. 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

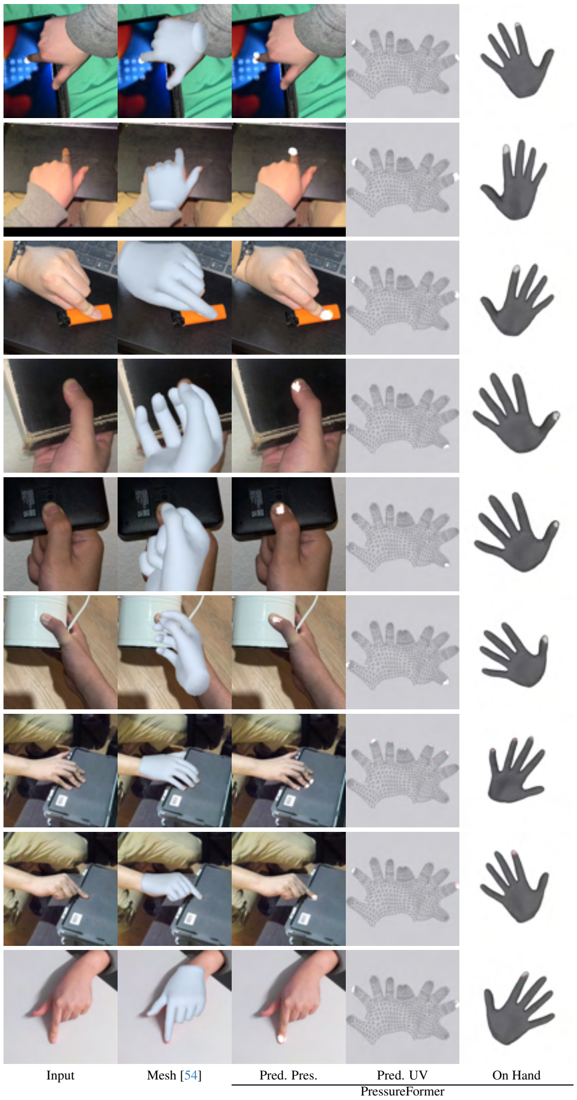


Figure S5. **Qualitative evaluation of PressureFormer on diverse, real-world examples featuring various objects and scenes.** Despite being trained exclusively on EgoPressure, the model recognizes pressure regions during corresponding contact events, demonstrating its potential for generalization.

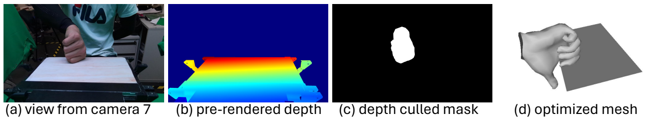

(a) view from camera 7 (b) pre-rendered depth (c) depth culled mask (d) optimized mesh

Figure S6. **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 $\mathcal{L}_{\vec{\mathbf{n}}}$	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] ↓	**5.68**	7.66	8.45	7.99	8.61	8.49	8.39	8.28

Table S2. **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. S1), depth volumetric IoU loss $\mathcal{L}_D$ (Eq. S1), mesh intersection loss $\mathcal{L}_{insec}$ (Sec. S2.1.2), as-rigid-as-possible loss $\mathcal{L}_{arap}$ (Sec. S2.1.2), Laplacian smoothness $\mathcal{L}_{lap}$ (Sec. S2.1.2), normal consistency regularization $\mathcal{L}_{\vec{\mathbf{n}}}$ (Sec. S2.1.2), and vertex offset regularization $\mathcal{L}_{offset}$ (Sec. S2.1.2). We demonstrate that each loss term contributes to our optimization performance.

initial tests, we noticed that this depth culling encourages 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 S6.

S2.1.4. 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:

$$\mathcal{L}_{\text{temp}} = \frac{1}{B-2} \sum_{i=1}^{B-2} \left\| \mathbf{J}_{i+2} - 2\mathbf{J}_{i+1} + \mathbf{J}_i \right\|_2 . \quad \text{(S4)}$$

S2.2. Evaluation of Annotation Fidelity

S2.2.1. 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 S2, 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 S7.

S2.2.2. 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

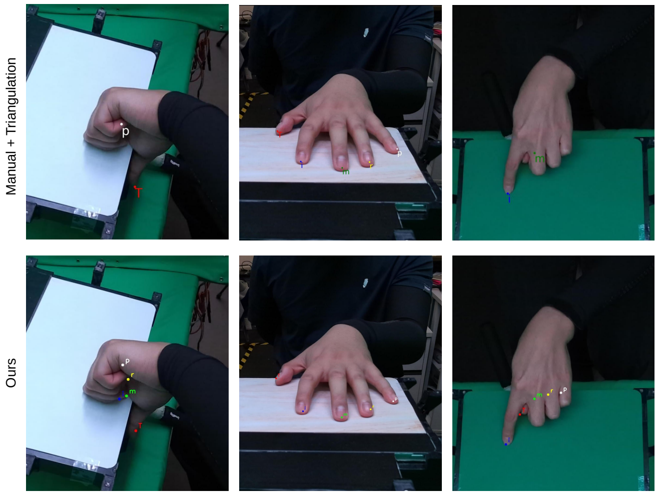


Figure S7. **Manual Verification Examples**. We demonstrate our annotation is accurate compared to the manual annotations. **(above)** We re-project the triangulated nail tips. We only triangulated them when they are visible in at least 2 views. **(bottom)** We re-project our 3D annotations which also show invisible nail tips as well.

for our dataset. As shown in Figure S9, although hand poses from HaMeR [26] appear plausible from a top view, side 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 [26] 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 S3. Additionally, Figure S10 presents a qualitative comparison of pressure estimation incorporating additional poses from HaMeR [26] 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 S4 and in Figure S8. 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 [26]	12.37	6.3

Table S3. **Hand pose verification.** Triangulation is performed on the nail tips using HaMeR [26] predictions across all static cameras, compared against manual annotations.

	MPJPE [mm]	Reconstruction Error [mm]
Finetuned HaMeR [26]	10.75	6.10
HaMeR [26]	18.58	8.11

Table S4. **Fine-tuning results** of HaMeR [26] on EgoPressure demonstrate improved hand pose accuracy, underscoring the value of our dataset for 3D hand pose estimation.

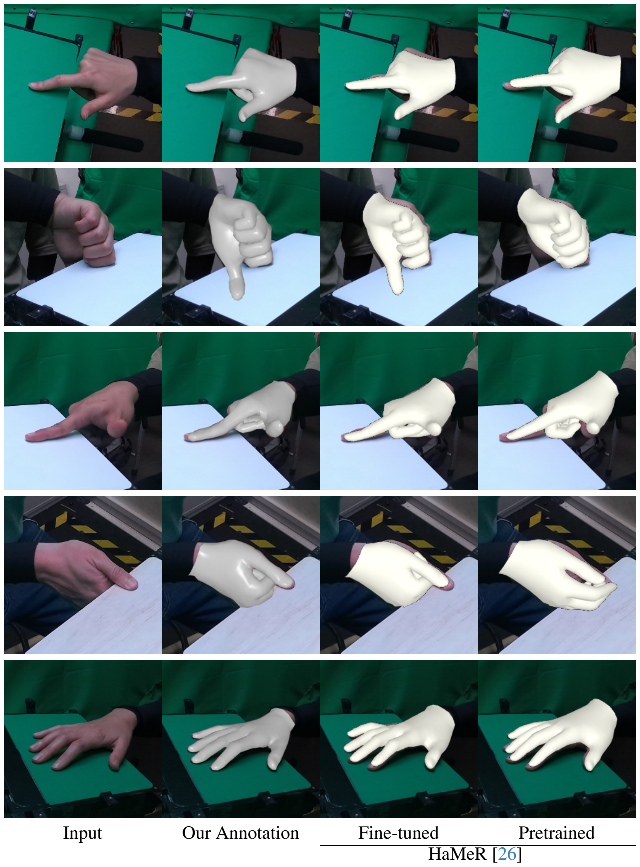


Figure S8. **Hand pose prediction and ground truth pose visualization for each camera.** We fine-tune HaMeR [26] on our dataset, demonstrating improved detail in hand pose estimation, particularly in scenarios where the hand interacts with a surface.

S3. Extended Details about Dataset

S3.1. Details about Gesture Description

Table S5 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.

S3.2. 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.

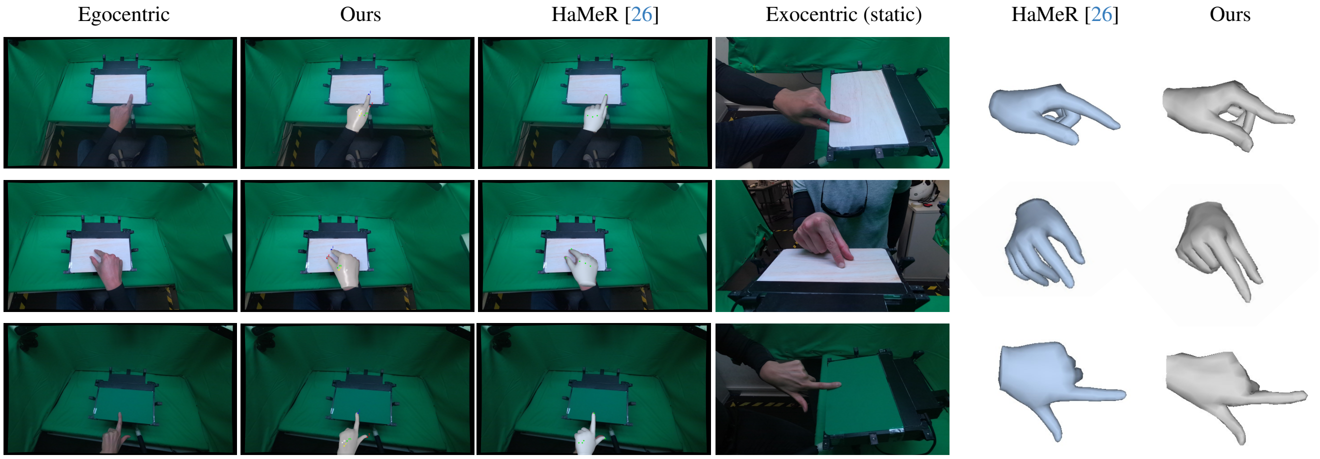


Figure S9. **Comparison of the estimated hand mesh from HaMeR [26] and our annotation method in both egocentric and exocentric views.** While the projected hand mesh from HaMeR appears visually plausible from an egocentric perspective, observable differences in hand articulation and mesh deformations become apparent from the exocentric viewpoint of the static cameras.

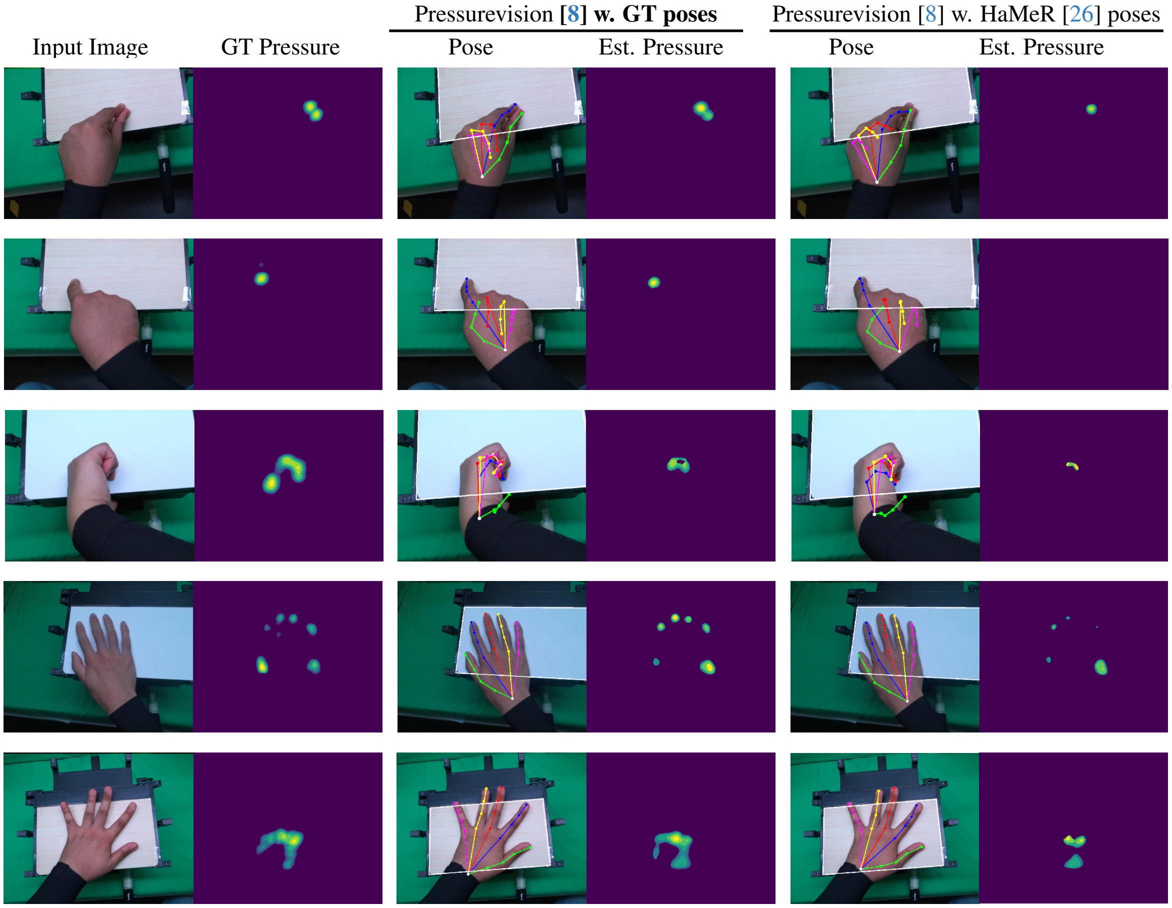
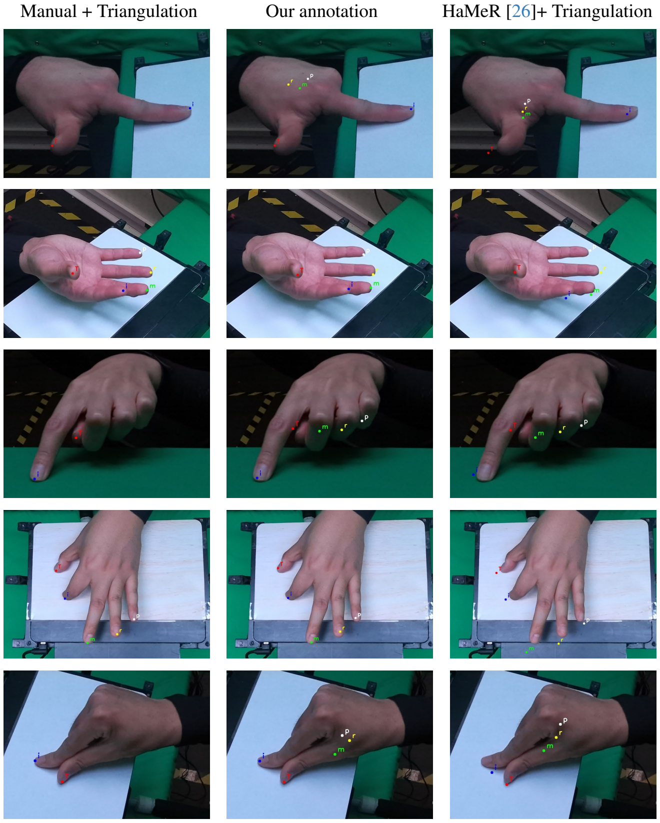

Pressurevision [8] w. GT poses
Pressurevision [8] w. HaMeR [26] poses
Input Image
GT Pressure
Pose
Est. Pressure
Pose
Est. Pressure

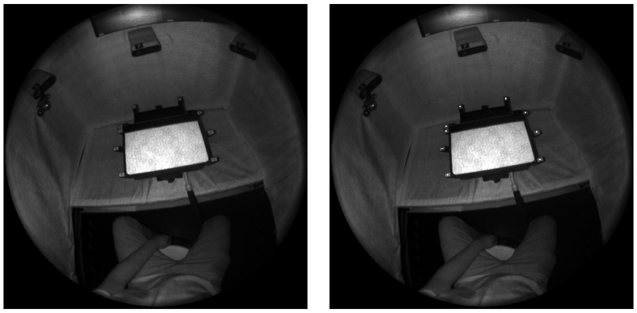
Figure S10. **Qualitative results of the image-projected baselines on egocentric views, incorporating additional hand pose inputs using our annotations and predictions from HaMeR [73].** We also reproject the area of the touchpad (indicated by white lines) to verify the egocentric camera pose.


Manual + Triangulation

Our annotation

HaMeR [26]+ Triangulation

Figure S11. **Qualitative comparison of reprojected nail tips** from our annotation method (**center**) and triangulation of HaMeR [26] predictions (**right**). The **left** column displays the reprojection of triangulated manually annotated visible tips.


Marker deactivated

Marker activated

Figure S12. **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 S5. **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.

S3.3. Dataset Comparisons

Table S6 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 [9]	2.9M	51	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
PressureVisionDB [8]	3.0M	36	✗	✗	✓	✓	✗	✓	✓	✗	Pressure sensor	✓	✗
ContactPose [1]	3.0M	50	✓	✓	✓	✓	✗	✓	✓	✓	Thermal imprint	✗	✗
GRAB [28]	1.6M	10	✓	✓	✗	✓	✗	✗	✗	✗	Inferred from Pose	✗	✗
ARCTIC [6]	2.1M	10	✓	✓	✗	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
H2O [21]	571k	4	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
OakInk [30]	230k	12	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
OakInk-2 [31]	4.01M	9	✓	✓	✓	✓	✓	✓	✓	✗	Inferred from Pose	✗	✗
DexYCB [2]	582k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
HO-3D [12]	103k	10	✓	✓	✓	✓	✗	✓	✓	✓	Inferred from Pose	✗	✗
TACO [24]	5.2M	14	✓	✓	✓	✓	✓	✓	✓	✓	Inferred from Pose	✗	✗
Affordpose [18]	26.7k	-	✓	✓	✓	✗	✗	✓	✗	✗	Inferred from Pose	✗	✗
AssemblyHands [25]	3.03M	34	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗
ContactArt [33]	332k	-	✓	✓	✓	✗	✗	✓	✓	✓	Simulated Pose	✗	✗
HOI4D [23]	2.4M	9	✓	✓	✓	✓	✓	✗	✓	✓	Inferred from Pose	✗	✗
YCBAfford [3]	133k	-	✓	✓	✓	✗	✗	✗	✗	✗	Simulated Pose	✗	✗
ObMan [14]	154k	-	✓	✓	✓	✗	✗	✗	✓	✓	Simulated Pose	✗	✗
FPHAB [7]	100k	6	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗
HA-ViD [32]	1.5M	30	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗
Ego4d [10]	3670 hours	923	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗
EPIC-KITCHEN-100 [4]	20M	37	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗
Ego-Exo4D [11]	1422 hours	740	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗

Table S6. **Comparison between EgoPressure and extended list of hand-contact datasets.**

S3.4. 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 S13) 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 S1. The effectiveness of this layout in facilitating accurate marker identification and pose estimation is further illustrated in Figure S14, 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.

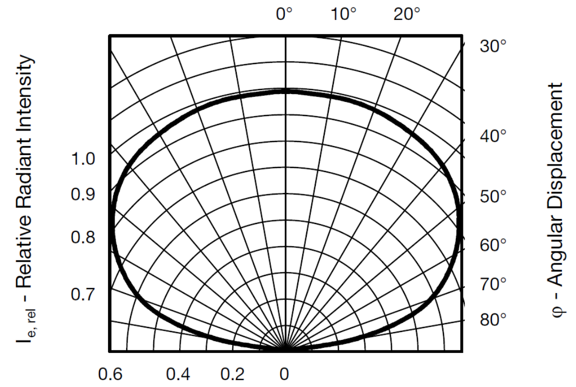


Figure S13. **Relative Radiant Intensity vs. Angular Displacement.** The marker enable a good visible radiant intensity of beam angle till 150 degree, which ensures good visibility in egocentric infrared camera.

S3.5. 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

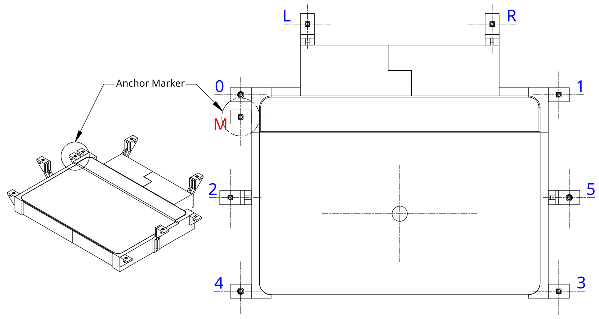


Figure S14. **Layout of the Active Markers.** The indices of the markers are aligned with the pseudocode provided in Algorithm S1. Starting from the asymmetrical anchor marker **M**, all markers can be identified by computing their relative distances and considering their spatial relationships.

Algorithm S1 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 S12) 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 S16).

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.

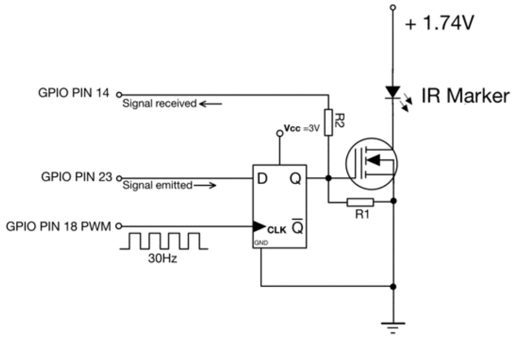


Figure S15. **Basic Electrical Elements Implementation.** We use a D-type flip-flop and N-channel MOSFET to ensure the IR marker will be activate by the next beginning of exposure after receiving signal from PIN 23. And PIN 14 will monitor the activation to obtain its timestamp.

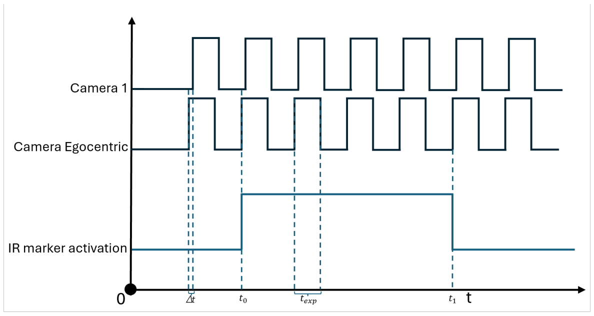


Figure S16. **Synchronization Diagram** We set head-mounted egocentric camera to align with 30 Hz triggering signal emitted by the Raspberry Pi CM4, this signal will also go to PIN 18 as clock frequency of D-type Flip-flop Fig. S15). Then exposure t_{exp} of all cameras is same. The other static cameras 1 to 7 will have a delay Δ 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.

S4. 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 [1]. However, we anticipate that our annotation method will extend naturally to more complex objects and interactions as these challenges are addressed. PressureVision++ [9] 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 S5.

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 S17) and has been successfully demonstrated to enhance commercial in-the-wild hand tracking [13, 34].

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.

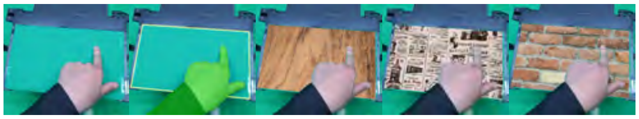
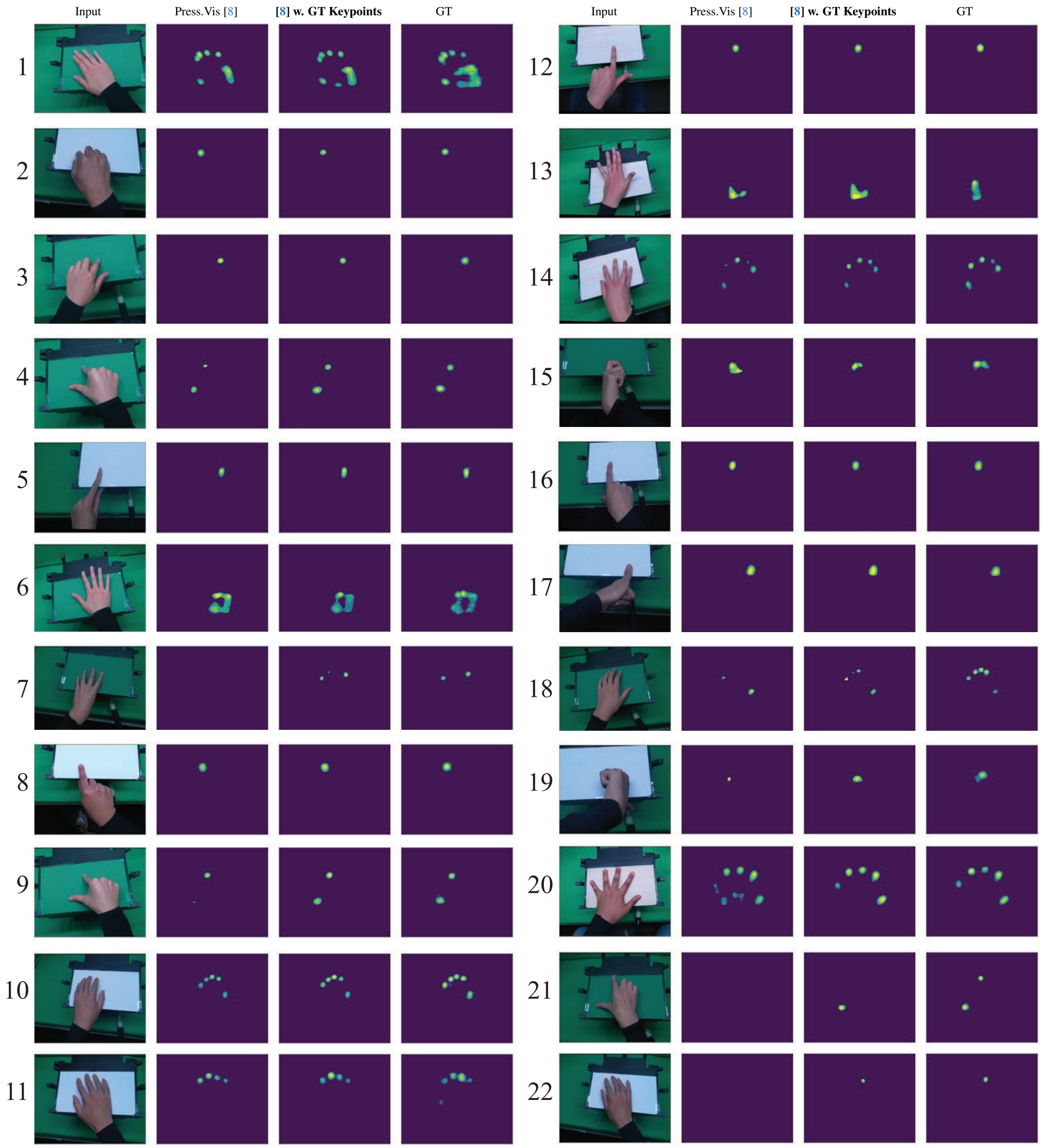


Figure S17. **Examples of background augmentation using hand masks and a touchpad.**

S5. Ethical Considerations

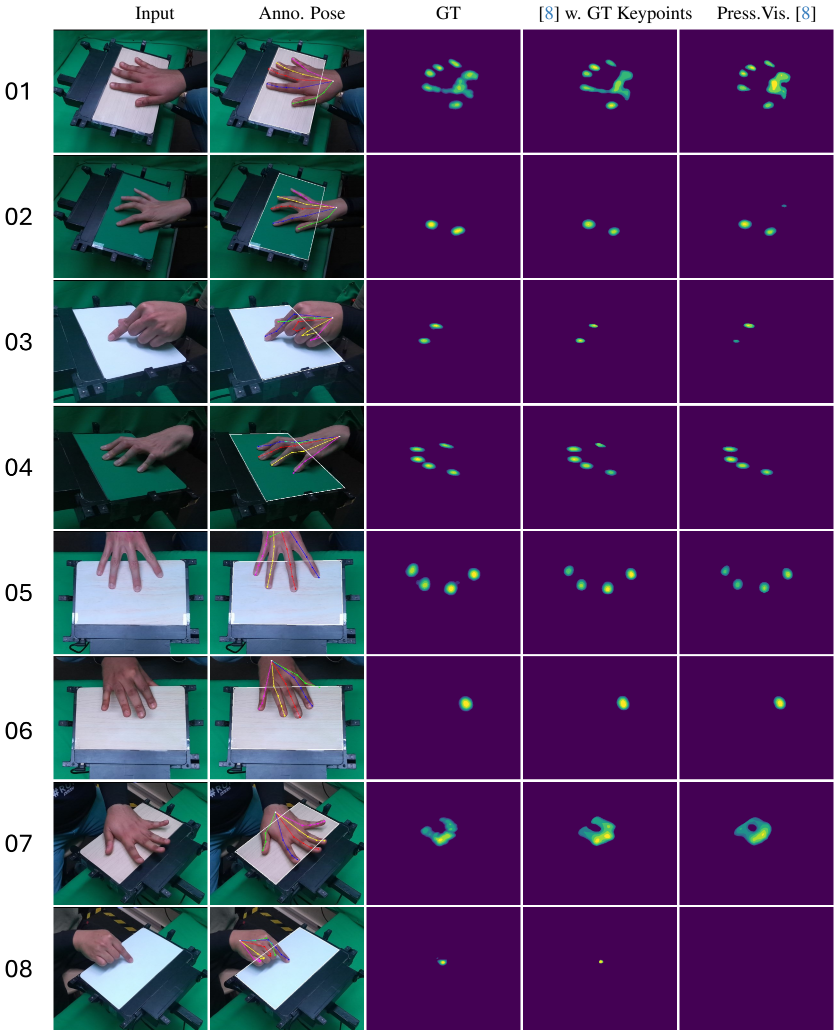
The recording and use of human activity data involve important ethical considerations. The EgoPressure project has received approval from ETH Zürich Ethics Commission as proposal EK 2023-N-228. This approval includes both the data collection and the public release of the dataset. All participants provided 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. [8]**	129.6	8.2	14.3	25.3	6.7	61.9	9.3	8.3	23.9	39.1	37.1
	[8] 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. [8]**	46.9	68.4	57.4	39.0	84.0	63.8	0.0	76.9	35.1	64.1	56.8
	[8] 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

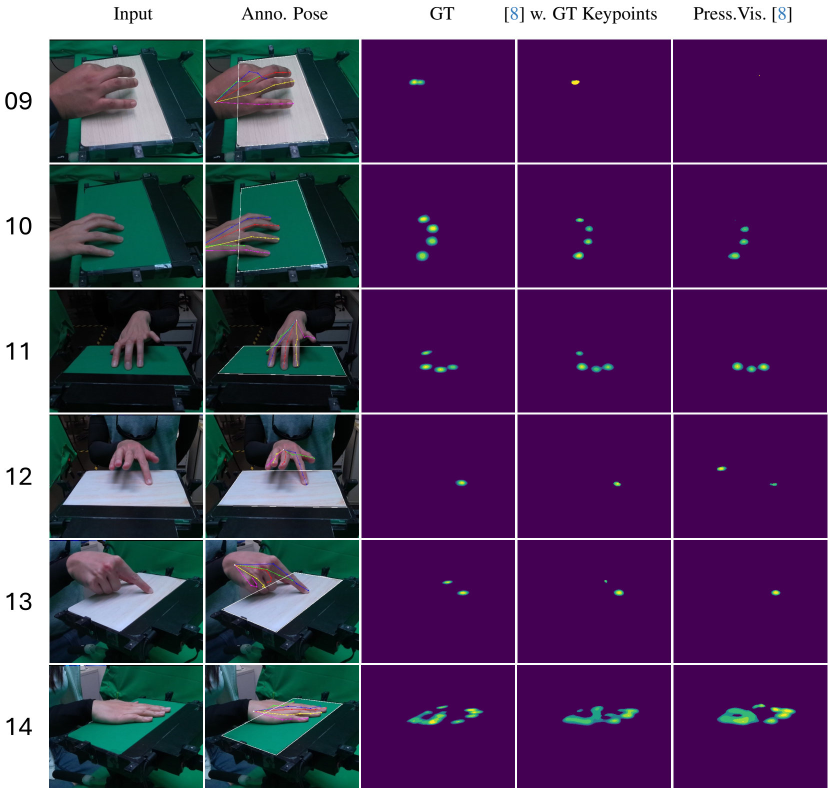
		12	13	14	15	16	17	18	19	20	21	22
MAE ↓	**Press.Vis. [8]**	6.2	50.1	39.4	54.7	11.7	10.7	32.0	37.0	67.4	21.5	8.29
	[8] w. GT Keypoints	4.7	51.7	35.6	30.4	10.0	10.3	24.9	32.6	35.5	14.7	4.89
Contact IoU ↑	**Press.Vis. [8]**	86.2	38.4	54.2	24.2	79.9	76.6	17.6	1.2	54.1	0.0	0.0
	[8] w. GT Keypoints	86.3	48.4	61.7	43.8	83.1	79.6	30.3	35.2	76.3	42.2	43.6

Figure S18. **Qualitative comparison of pressure maps** inferred using PressureVisionNet [8] 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.



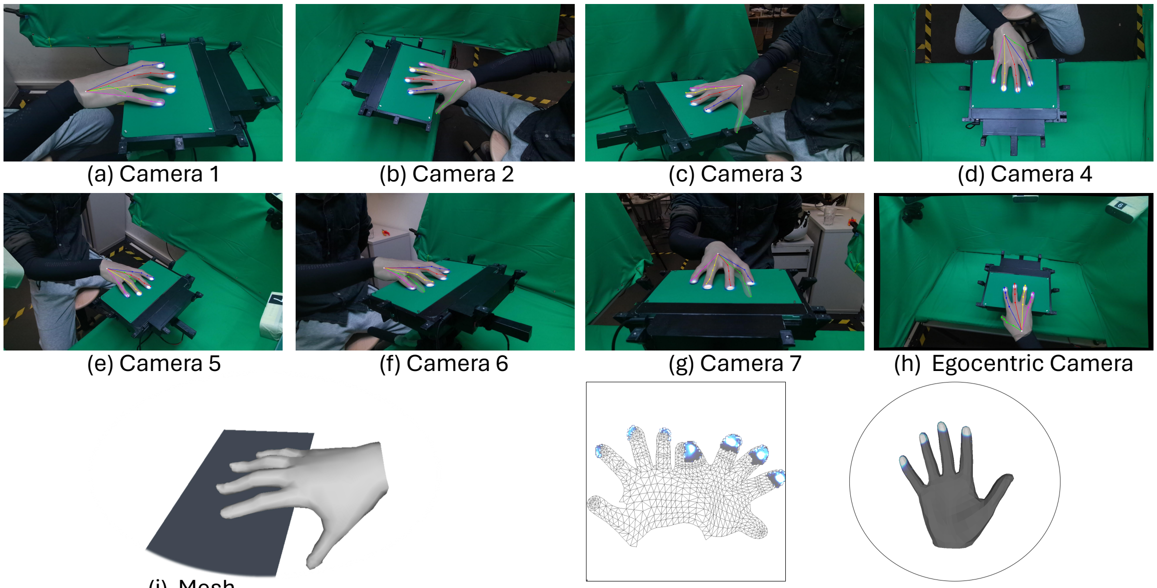
		01	02	03	04	05	06	07	08
MAE [Pa] ↓	**Press.Vis. [8]**	136.5	114.8	22.3	57.1	56.4	51.3	71.6	22.1
	[8] w. GT Keypoints	117.0	106.2	20.5	50.5	49.9	46.4	50.6	9.1
Contact IoU [%] ↑	**Press.Vis. [8]**	65.6	74.9	51.9	63.1	59.2	77.8	59.5	0.0
	[8] w. GT Keypoints	78.1	77.4	58.1	69.2	65.1	83.2	67.7	31.1
	Camera #	2	2	3	3	4	4	5	5

Figure S19. **Comparison of pressure maps** estimated by PressureVisionNet [8] 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. [8]**	52.8	77.4	119.5	76.5	62.6	199.6
	[8] w. GT keypoints	23.4	70.8	102.7	43.7	52.8	182.0
Contact IoU ↑	**Press.Vis. [8]**	0.0	43.6	60.1	17.4	48.2	49.3
	[8] w. GT keypoints	38.5	56.0	68.9	43.5	55.7	57.1
Camera #		1	1	7	7	6	6

Figure S20. **Comparison of pressure maps** estimated by PressureVisionNet [8] and our adapted model, evaluated using input images from cameras 1, 6, and 7. The models are the same as in Figure S19, which are trained on images from camera views 2, 3, 4, and 5.

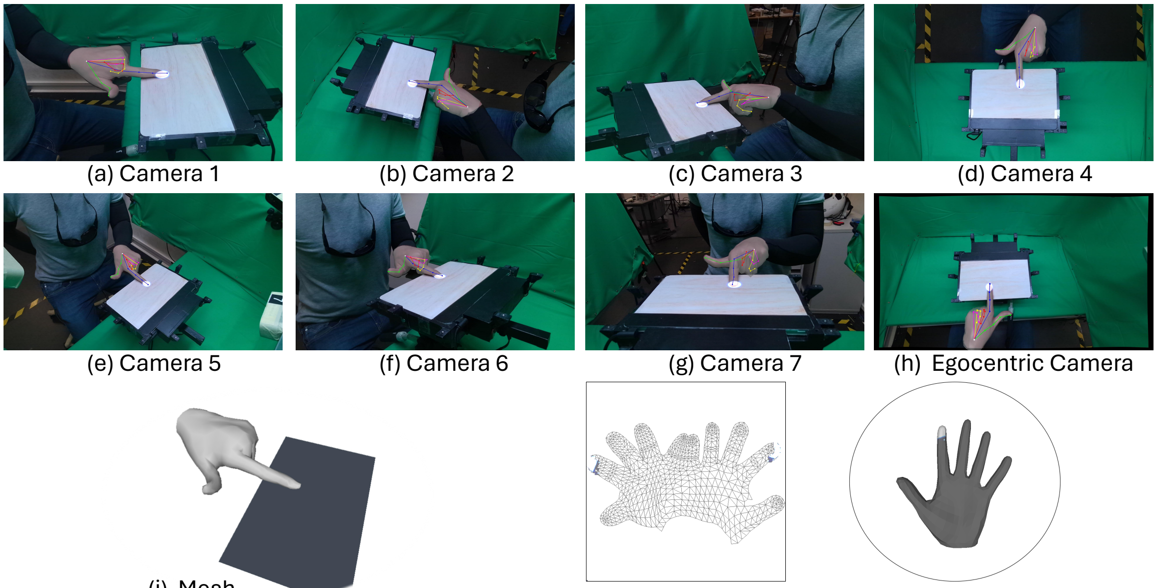


(a) Camera 1 (b) Camera 2 (c) Camera 3 (d) Camera 4

(e) Camera 5 (f) Camera 6 (g) Camera 7 (h) Egocentric Camera

(i) Mesh (j) Pressure texture (k) Pressure on hand

Figure S21. **Example of Annotation 1.** Right hand with gesture: grasp edge with uncurled thumb down.



(a) Camera 1 (b) Camera 2 (c) Camera 3 (d) Camera 4

(e) Camera 5 (f) Camera 6 (g) Camera 7 (h) Egocentric Camera

(i) Mesh (j) Pressure texture (k) Pressure on hand

Figure S22. **Example of Annotation 2.** Left hand with gesture: index press with high force.

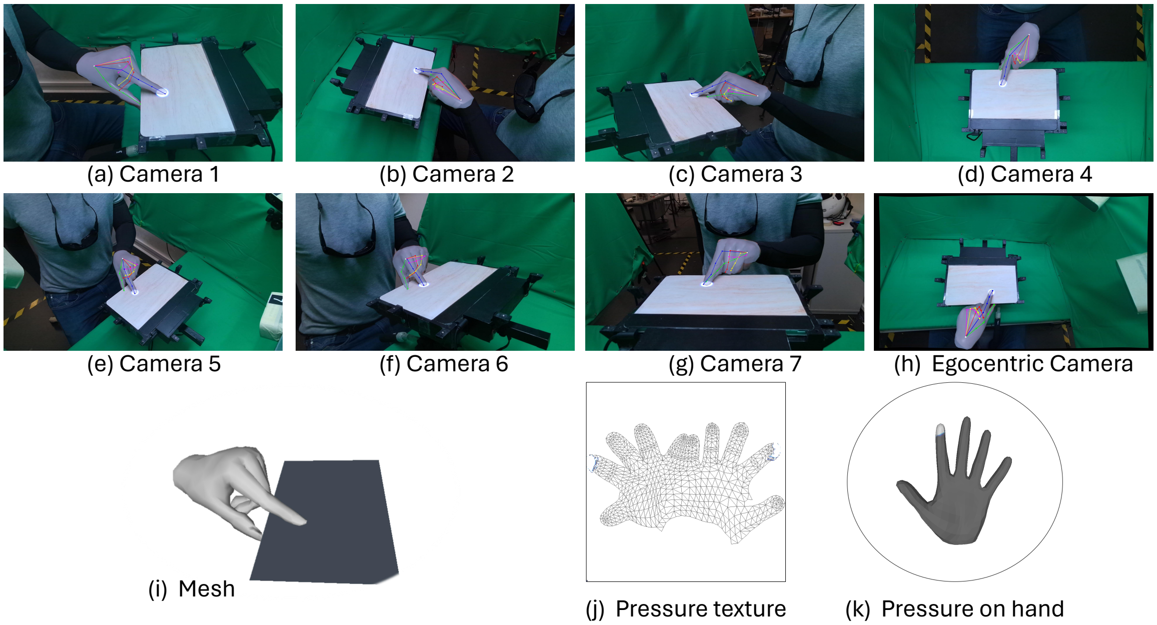


Figure S23. **Example of Annotation 3.** Left hand with gesture: pinch thumb down on the edge with high force.

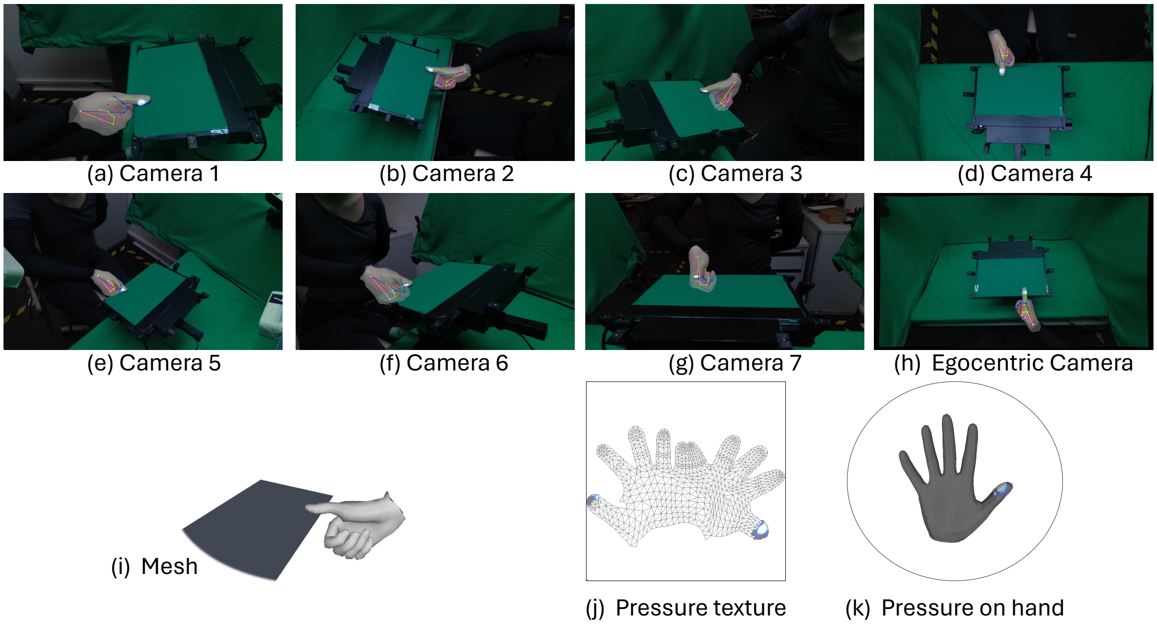


Figure S24. **Example of Annotation 4.** Right hand with gesture: grasp edge with curled thumb up.

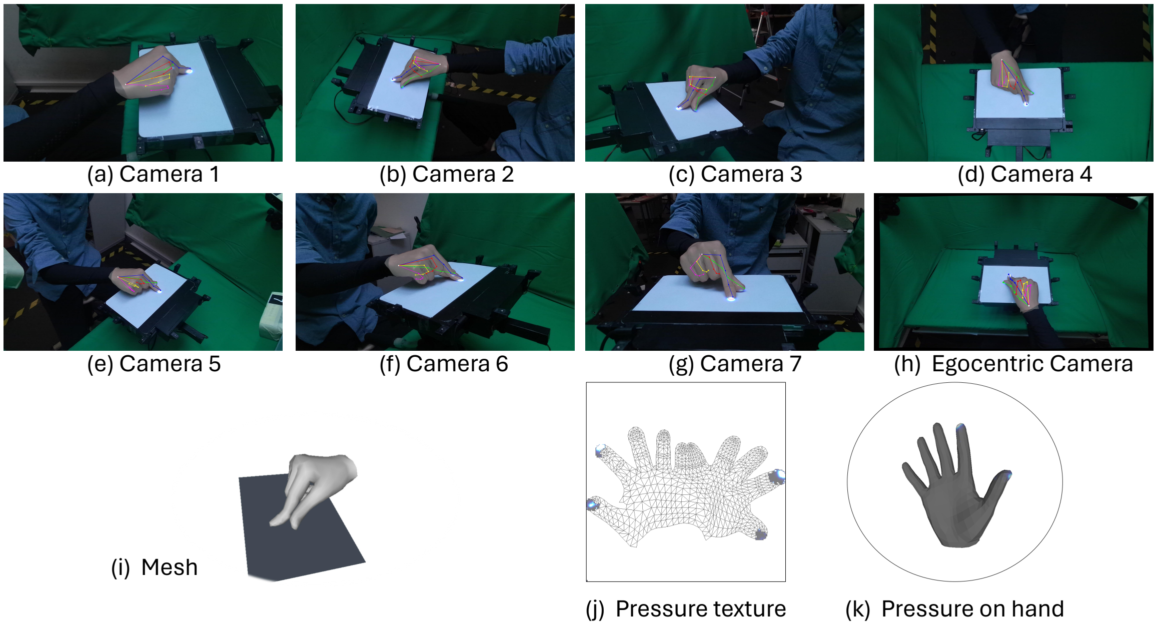


Figure S25. **Example of Annotation 5.** Right hand with gesture: pinch finger zoom in and out.

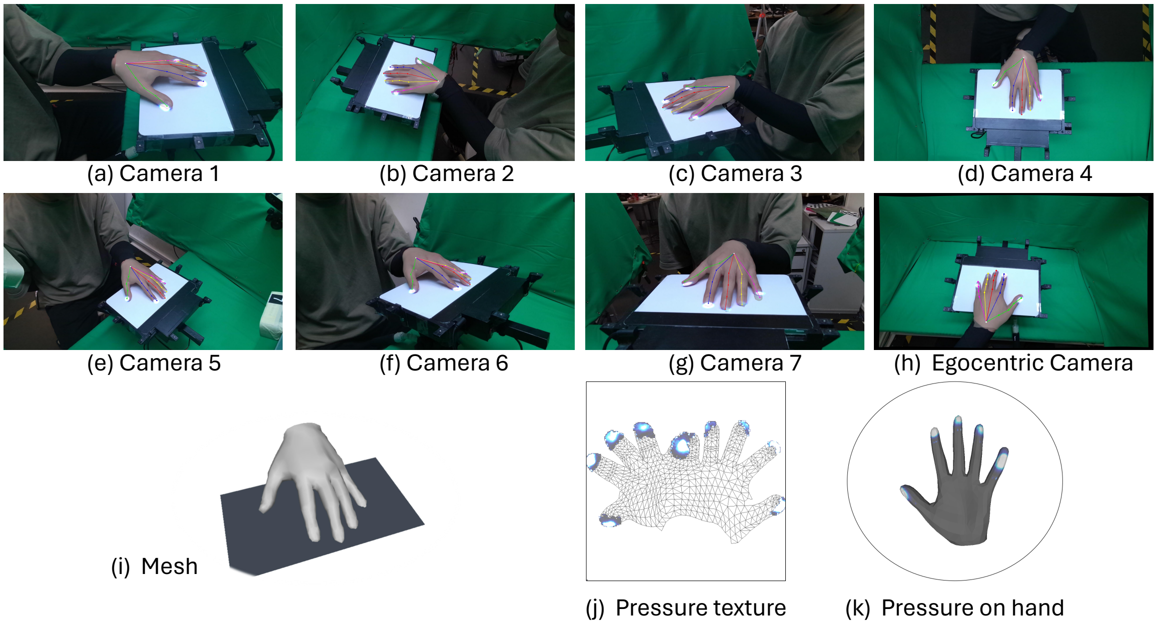


Figure S26. **Example of Annotation 6.** Left hand with gesture: pull all fingers towards the participant.

References

[1] 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. 8, 10

[2] 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. 8

[3] 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. 8

[4] 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

[5] 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

[6] 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. 8

[7] 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. 8

[8] 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, 8, 11, 12, 13

[9] 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, 8, 10

[10] 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. 8

[11] 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. 8

[12] 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. 8

[13] 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. 10

[14] 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. 8

[15] 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

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

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

[18] 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

[19] 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. 3

[20] 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. 2

[21] 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. 8

[22] 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

[23] 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. 8

[24] 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. 8

[25] 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

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

[27] 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

[28] 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. 8

[29] 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. 3

[30] 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. 8

[31] 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. 8

[32] 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

[33] 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. 8

[34] 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. 10