Transformer-based Unified Recognition of Two Hands Manipulating Objects

Hoseong Cho Chanwoo Kim Jihyeon Kim Seongyeong Lee
Elkhan Ismayilzada Seungryul Baek

UNIST, South Korea

Abstract

Understanding the hand-object interactions from an egocentric video has received a great attention recently. So far, most approaches are based on the convolutional neural network (CNN) features combined with the temporal encoding via the long short-term memory (LSTM) or graph convolution network (GCN) to provide the unified understanding of two hands, an object and their interactions. In this paper, we propose the Transformer-based unified framework that provides better understanding of two hands manipulating objects. In our framework, we insert the whole image depicting two hands, an object and their interactions as input and jointly estimate 3 information from each frame: poses of two hands, pose of an object and object types. Afterwards, the action class defined by the hand-object interactions is predicted from the entire video based on the estimated information combined with the contact map that encodes the interaction between two hands and an object. Experiments are conducted on H2O and FPHA benchmark datasets and we demonstrated the superiority of our method achieving the state-of-the-art accuracy. Ablative studies further demonstrate the effectiveness of each proposed module.

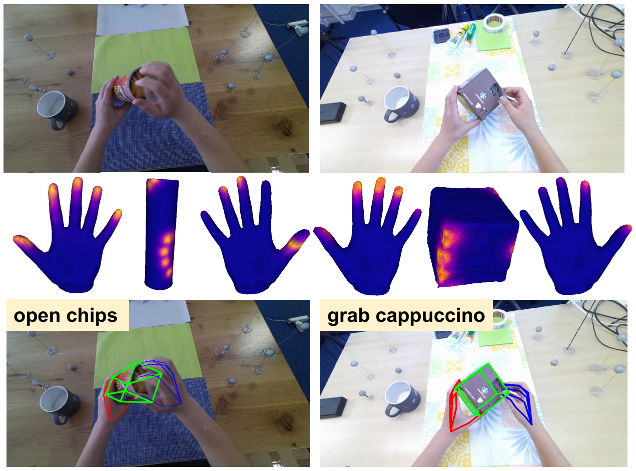


Figure 1. Example results of pose estimation and interaction recognition for two hands manipulating objects. Our method first estimates hand poses, object poses and object types. Then, interaction class is estimated using estimated information combined with contact maps. (Row 1) example input video $\mathbf{v}$ for *open chips* and *grap cappuccino*; (Row 2) contact maps for left hand $\mathbf{m}^{\text{Left}}$, object $\mathbf{m}^{\text{O}}$, and right hand $\mathbf{m}^{\text{Right}}$; (Row 3) estimated 3D poses of hands $\mathbf{h}$, a 3D object pose $\mathbf{o}$ and the estimated interaction class $\mathbf{a}$.

1. Introduction

Estimating poses and actions of an egocentric video involving two hands and an object is an important factor of various applications such as augmented reality (AR), virtual reality (VR) and human computer interaction (HCI). Previously, there has been much progress in the hand pose estimation [3–5,11,12,18,31,33,38,43,53,61] and in the object 6D pose estimation [10,26,28,36,51,57,58] separately from each other. Recently, there has been a surge in demand for understanding hand-object interactions, leading to the emergence of methods for joint pose estimation of hands and objects [22,23,39]. However, most methods focus on the separate problem either for the pose estimation [9,13,20,39] or for the interaction recognition [6,42,48]. Furthermore, most approaches developed the pose estimation method based on the already cropped tight bounding boxes of hands and objects which are not realistic. Therefore, the pose estimation accuracy is frequently affected by the performance of the detector.

To tackle the issue, Tekin et al. [50] proposed an unified framework that estimates the 3D hand pose, the object 6D pose and their action classes. They developed the pose estimator extending the architecture of [45] towards 3D space and recognize actions using estimated hand and object poses. The long short-term memory (LSTM) [25]-based architecture is further used to map the information towards the action classes. Kwon et al. [32] further extended the framework towards involving two hands rather than one hand: They estimated 3D poses of two hands, 6D pose of an object and their action classes. The proposed method involves the graph convolutional network (GCN) to model the hand-object interaction considering the geometric relation between hand and object. In both works, estimated hand and object poses (i.e. skeletons) were used as the cue to the interaction recognition.

In this paper, we propose the Transformer-based unified framework (H2OTR) to estimate poses of two hands, object pose, object types and interaction classes between hands and object. We construct the Transformer-based architecture similarly to [7, 60] and it is able to predict the poses from each frame without hand/object detectors or any additional post-processing such as non-maximal suppression (NMS). It also estimates hand-object interaction classes from the entire videos. We additionally exploit the contact map between hand and object meshes by recovering hand meshes from hand poses via inverse kinematics. We demonstrated that the contact map expresses the explicit relational information between hands and object and is used as the crucial cue for the hand-object interaction recognition task. We summarize our contributions in this paper as follows:

- We propose the Transformer-based unified framework for estimating poses of two hands, object poses, object types and hand-object interaction classes at a single inference step.
- We introduce a novel interaction recognition method which utilizes a contact map. To the best of our knowledge, this is the first work to exploit the contact map as a cue for interaction recognition.
- We achieve the state-of-the-art performance in pose estimation and interaction recognition tasks using H2O [32] and FPHA [18] datasets.

2. Related work

In this section, we introduce related researches about our work: 1) Hand and object pose estimation from monocular RGB images, 2) Hand-object interaction recognition, 3) Transformer in vision tasks.

2.1. Hand and object pose estimation

3D Hand pose estimation. Hand pose estimation has been studied extensively in recent years. Traditionally, 3D hand pose estimation has been mainly done in a depth image using the Kinect sensor. Depth-based hand pose estimation generally utilizes datasets that can handle various camera perspectives, pose variations and shape. However, annotating such datasets is very expensive. In order to solve this problem, Baek et al. [3] proposed a method of synthesizing data in skeleton space. They synthesized depth map entries by utilizing hand pose generator which is learned to synthesize depth maps from skeleton entries. Hand pose estimation from a single RGB image are also being steadily appearing [4, 5, 12, 31, 38, 61]. Zimmerman et al. [61] first proposed to estimate the 3D hand pose on regular RGB images using deep CNNs. Recently, Lin et al. [38] proposed a graph convolution reinforced Transformer methodology that estimates pose and mesh on a single image. More recently, related research has been expanded to the 3D hand pose estimation based on various viewpoints and video inputs [11, 18, 53]. Garcia-Hernando et al. [18] collected RGB-D video, and estimated 3D hand pose from the video frame. Chen et al. [11] estimated 3D hand poses with the self-supervised learning methodology without involving explicit 3D annotations.

Object 6D pose estimation. The main idea of object 6D pose estimation is to estimate the 6 degrees of freedom (6-DoF) position and orientation of rigid objects in 3D space. Object pose estimation has been actively studied until recently, and they are performed on both depth and single RGB images. Most recent methods [10, 36, 57] directly regress object poses using CNNs to map the observed image into the 3D object poses. Representatively, Chen et al. [10] proposed a model for estimating category-level 6D object size and pose. They learned canonical shape space to solve the intra-class shape variation issue. However, direct pose estimation is affected by the occlusion driven from other obstructions or various lighting. To relieve the issue, correspondence-based method has been proposed recently. Tekin et al. [51] presented a single-shot method for detecting the position of a object in a single RGB image and estimating a 6D pose. In addition, methods for performing pose refinement based on roughly estimated initial poses were also shown in multiple literatures [26, 28, 58].

Hand-object pose estimation. Recently, studies to understand the interaction between hands and objects [2, 13, 27, 39] have been active. Hasson et al. [23] first created a synthetic dataset ”Obman” in which hands and objects interact with, and proposed a methodology for reconstructing hands and objects at the same time. Afterwards, real datasets [9, 18, 20] involving hand-object interaction began to emerge, methodologies for estimating the object 6D pose and 3D hand pose were proposed at the same time. However, in the real-world scenario, people mostly interact with objects with two-hands, whereas the existing datasets and methods only consider the single-hand cases. Then, Kwon et al. [32] proposed the H2O dataset that captured the scenario where two-hands and an object are interacting each other, and proposed a method for estimating the poses of two-hands and an object simultaneously based on [45].

2.2. Hand-object interaction recognition

Action recognition is a classic vision task and it is mainly performed using the RGB-based features such as 3Dconv [52], I3D [8], Two-stream [47] and SlowFast [17]. Earlier works [6, 15, 16, 42, 44, 48] on the hand-object interaction recognition task adopted similar appearance cues to recognize the hand-object interaction in the egocentric viewpoint. Recently, Garcia-Hernando et al. [18] showed that 2D and 3D hand poses are more helpful than RGB-based features in recognizing hand actions. Tekin et al. [50] and Kwon et al. [32] proposed an unified framework that performs pose estimation of hands and objects and the hand-object interaction recognition via long short-term memory (LSTM) or graph convolutional network (GCN), respectively. In this paper, we present the Transformer-based unified framework for the same task.

2.3. Transformers in vision

Transformer showed excellent performance in natural language field. It has also been extended to the field of computer vision and has been successfully applied to various tasks such as object segmentation, detection, and pose estimation. Dosovitskiy et al. [14] solved the long-range dependency problem between the pixels of CNN architecture by dividing the image into patches. So, they achieved the state-of-the-art performance in the image classification task. Lin et al. [37,38] and Hampali et al. [21] used Transformer architecture for pose reconstruction and estimation. Carion et al. [7] proposed the Transformer-based architecture that deals with the object detection task without additional processing such as NMS. However, they had problems with high computational cost and slow convergence time. To relieve the issue, many studies [34, 40, 55, 59, 60] have been appeared. Especially, Zhu et al. [60] predicted the reference point for each object query of the decoder and performed cross-attention only with the offset points of the reference point. As a result, they improved computational cost and slow convergence speed. Human-object interaction (HOI) is a task that predicts human and object bounding boxes and their interactions. Several methods [29,30,49] have demonstrated the state-of-the-art performance by utilizing the above-mentioned advantages. Carion et al. [7]'s work has been recently applied to a multi-human pose estimation task [41, 62]. Unlike the top-down or bottom-up methods, they could estimate robust poses without additional post-processing. In this paper, we dealt with the similar problem in the two hands manipulating object scenario which involves its own challenges different from object detection or human pose estimation tasks.

3. Transformer-based unified framework for two hands and an object: H2OTR

We propose the Transformer-based unified framework, called H2OTR $f^{\text{H2OTR}}: V \rightarrow [H,O,C,A]$ that simultaneously estimates outputs of 4 tasks from a given video $\mathbf{v} \in V \subset \mathbb{R}^{N_V \times W \times H \times 3}$: hand poses $\mathbf{h} \in H \subset \mathbb{R}^{N_V \times 2 \times 21 \times 3}$, object poses $\mathbf{o} \in O \subset \mathbb{R}^{N_V \times 21 \times 3}$, probability of object types $\mathbf{c} \in C \subset \mathbb{R}^{N_V \times N_c}$, and probability of interaction classes $\mathbf{a} \in A \subset \mathbb{R}^{N_a \times 1}$, where N_V, W, H, N_c and N_a denote the number of frames in a video, width and height of the video frames, number of object-types and the number of interaction classes, respectively.

Our H2OTR $f^{\text{H2OTR}} = [f^{\text{HOP}}, f^{\text{IA}}]$ is divided into two sub-modules: the hand-object pose estimation network $f^{\text{HOP}}: V \rightarrow [H,O,C]$ that estimates hand poses H, object poses O and object types C from each frame of the video V and the hand-object interaction recognition network $f^{\text{IA}}: [H,O,C] \rightarrow A$ that performs the interaction recognition of the whole video V using the estimated hand poses H, object poses O and object types C as the input.

The hand pose vector $\mathbf{h}$ is defined as 21×3-dimensional array that indicates xyz-coordinates of 21 hand poses. The object pose vector $\mathbf{o}$ is defined as 21×3-dimensional array that consists of xyz-coordinates of 8 corners, 12 edge midpoints and the centroid of the 3D bounding box that tightly surrounds the objects. The object 6D pose is recovered in the hand-object interaction recognition network f^{IA} from $\mathbf{o}$ applying the rigid alignment [19] between the predicted object pose and the ground-truth object pose. The object type is defined based on the class of foreground objects such as 'chips', 'book', etc, combined with a 'background' class and it also include two more classes for 'left hand' and 'right hand'.

In the remainder of this section, we will investigate more details on each sub-module.

3.1. Hand-object pose estimation

The hand-object pose estimator $f^{\text{HOP}}: V \rightarrow [H,O,C]$ is involved at this stage, to estimate the poses of two hands, object poses and object types. It first extracts the multi-scale image features $\mathbf{s}_i \in S \subset \mathbb{R}^{\frac{H}{2^i} \times \frac{W}{2^i} \times d}$ from each frame $\mathbf{x} \in X \subset \mathbb{R}^{W \times H \times 3}$ contained in the overall video $\mathbf{v} \in V$ and input them into the encoder, decoder and prediction head of the Transformer architecture. We will elaborate each step in the remainder of this subsection.

Backbone. We employed the ImageNet pre-trained ResNet50 [24] architecture as our backbone network. Given each frame $\mathbf{x}$ of the overall video $\mathbf{v}$, the backbone network is applied to extract multi-scale feature maps, denoted as $\mathbf{s}_3$,$\mathbf{s}_4$ and $\mathbf{s}_5$. The spatial dimension of $\mathbf{s}_i$ is represented as $\frac{H}{2^i} \times \frac{W}{2^i}$. After that, to project the channel dimension of each feature map equally as d dimension, we use the 1×1 convolution layer as follows:

$$\mathbf{s}_i^{\text{proj}} = \text{Conv}_i(\mathbf{s}_i), \quad \mathbf{s}_i^{\text{proj}} \in \mathbb{R}^{\frac{H}{2^i} \times \frac{W}{2^i} \times d} \tag{1}$$

where $d = 256$, $i = 3,4,5$ is used and $\text{Conv}_i(\cdot)$ denotes the 1×1 convolution layer in each multi-scale level. We also obtain the additional feature map $\mathbf{s}_6^{\text{proj}} \in \mathbb{R}^{\frac{H}{64} \times \frac{W}{64} \times d}$ from $\mathbf{s}_5$ via convolution layer which has 3×3 kernels and stride 2. We flatten each feature map $\mathbf{s}_i^{\text{proj}}$ along the spatial-axis resulting in a vector $\mathbf{s}'_i \in \mathbb{R}^{\frac{H}{2^i} \cdot \frac{W}{2^i} \times d}$. The input to the Transformer encoder $\mathbf{s} \in \mathbb{R}^{\sum_{i=3}^{6} (\frac{H}{2^i} \cdot \frac{W}{2^i}) \times d}$ is obtained by adding 2D positional embedding $\mathbf{P}_i \in \mathbb{R}^{\frac{H}{2^i} \cdot \frac{W}{2^i} \times d}$ and level embedding $\mathbf{L}_i \in \mathbb{R}^{\frac{H}{2^i} \cdot \frac{W}{2^i} \times d}$ to the i-th scale feature map $\mathbf{s}'_i$ and concatenating responses in all scales as follows:

$$\mathbf{s} = \Big\Vert_{i=3}^{6} (\mathbf{s}'_i + \mathbf{P}_i + \mathbf{L}_i) \tag{2}$$

where $\Vert$ is the concatenation operation, level embedding $\mathbf{L}_i$ is learnable parameters to distinguish multi-scale feature maps and $\mathbf{P}_i$ is a sinusoidal positional embedding describing spatial locations.

Encoder. Since the scale of the hand and object varies greatly in the image space according to the distance from the camera, multi-scale feature maps may useful to predict instances with

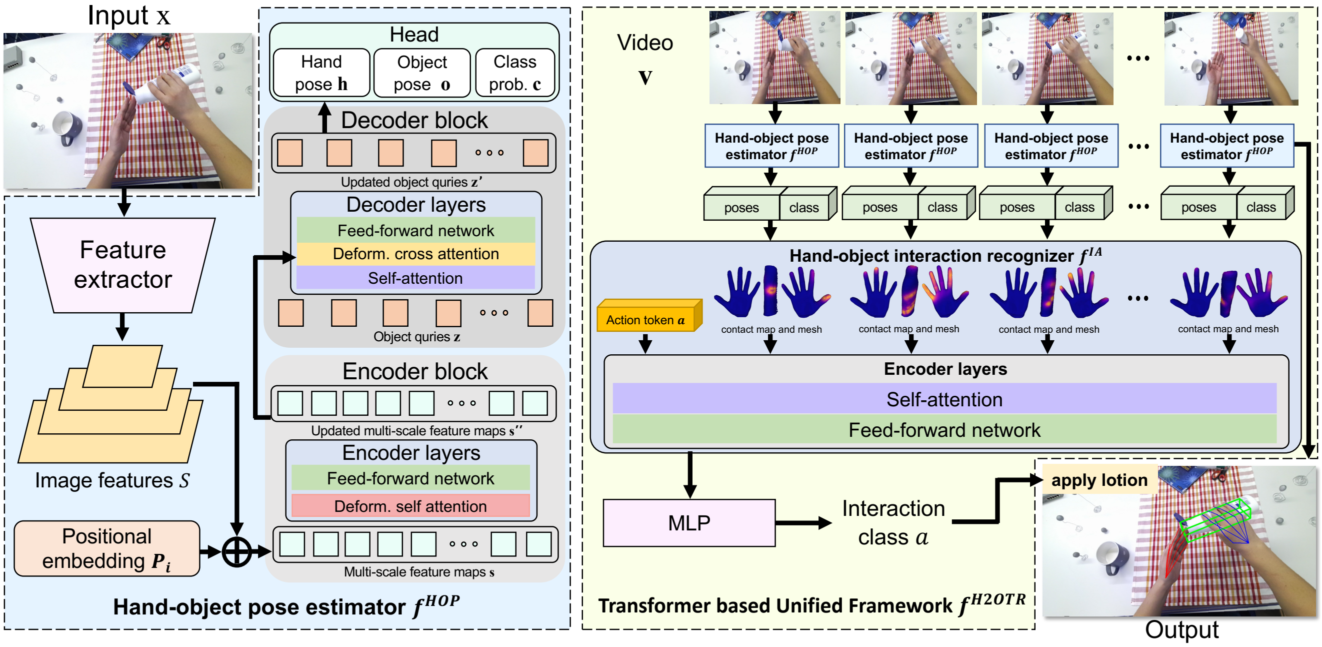


Figure 2. Schematic of overall framework. Our network consists of two-stages: First, the pose estimator f^{HOP} processes each frame $\mathbf{x}$ of a video $\mathbf{v}$ involving two hands manipulating objects. We extract multi-scale image features $\mathbf{s}_i$ using ResNet-50 [24] and pass them on the Transformer Encoder-Decoder Layer. The output of f^{HOP} is prediction for 3D hand poses $\mathbf{h}$, object poses $\mathbf{o}$ and classes $\mathbf{c}$. Second, the interaction recognizer f^{IA} receives the set of estimated 3D hand poses, object poses and classes combined with the contact map $\mathbf{m}$ to estimate the hand-object interaction classes $\mathbf{a}$.

various scales in our task. However, multi-scales exhibit higher computational cost than a single feature map; to relieve this, we use the deformable attention operation in the encoder layer as in [60]. The overall encoder is composed of a stack of 6 encoder layers, and each encoder layer consists of a multi-head deformable self-attention layer with 8 heads followed by a feed forward network. Given input features $\mathbf{s}$, the encoder extracts updated features $\mathbf{s}'' \in \mathbb{R}^{\sum_{i=3}^{6}(\frac{H}{2^i}\cdot\frac{W}{2^i})\times d}$ which are used in the deformable cross-attention operation of the decoder.

Decoder. The object queries $\mathbf{z} \in Z \subset \mathbb{R}^{N\times d}$ are learnable parameters which are randomly initialized. The decoder updates object queries to estimate the pose of each instance. The object queries extract features from the encoder output feature maps $\mathbf{s}''$ for deformable cross-attention. The reference point $\mathbf{r}\in\mathbb{R}^{2\times1}$ denotes the xy spatial locations specifying where to attend in the 2D spatial features for the object queries, and it is estimated from the object queries $\mathbf{z}$ via a fully-connected (FC) layer. The decoder is also a stack of 6 decoder layers, and each decoder layer consists of a self-attention layer, a multi-head deformable cross-attention layer having 8 heads [60] and a feed forward network. Finally, we get the updated output queries $\mathbf{z}' \in \mathbb{R}^{N\times d}$ from the decoder. The updated queries $\mathbf{z}'$ are fed into the prediction head to predict N sets of hand poses $\mathbf{h}\in\mathbb{R}^{21\times3}$, object poses $\mathbf{o}\in\mathbb{R}^{21\times3}$ and probability of object types $\mathbf{c}\in\mathbb{R}^{N_c\times1}$.

Prediction head. In the Transformer-based object detection framework [7, 60], the bounding box is predicted from the same head for all types of classes. Contrary to this, since the data distributions of hand poses and object poses are far different in our pipeline, we used separate head $f_{\text{hand}}: Z\rightarrow H$, $f_{\text{obj}}: Z\rightarrow O$ for hands and objects to predict two offsets $\Delta\mathbf{h}\in\mathbb{R}^{21\times3}$ and $\Delta\mathbf{o} \in \mathbb{R}^{21\times3}$ from $\mathbf{z}'$. The final hand and object poses are predicted by adding the offset to the reference points, as follows:

$$\mathbf{h}=\rho(\Delta\mathbf{h}+\rho^{-1}(\mathbf{r})), \quad \mathbf{o}=\rho(\Delta\mathbf{o}+\rho^{-1}(\mathbf{r})), \tag{3}$$

where ρ is a sigmoid function and $\mathbf{r}\in\mathbb{R}^{2\times1}$ is a reference point on the frame $\mathbf{x}$. In addition, class probabilities $\mathbf{c}\in\mathbb{R}^{N_c\times1}$ is predicted by another head $f_{\text{cls}}: Z\rightarrow C$ as follows:

$$\mathbf{c}=f_{\text{cls}}(\mathbf{z}') \tag{4}$$

Finally, we get the prediction set $\mathbf{y}=\{\mathbf{h},\mathbf{o},\mathbf{c}\}$. The predicted poses of hands and objects are in the UVD space and they are later converted into the camera space using the camera intrinsic parameters.

Reference point refinement. Since the reference point $\mathbf{r}$ is used to specify the location to sample the feature vector in the image, it is important to locate the reference points $\mathbf{r}$ at the position of the instances. Therefore, we used the mechanism to refine the reference points $\mathbf{r}$ to increase the performance, similar to [60]. The reference points are refined based on hand pose $\mathbf{h}$ and object pose $\mathbf{o}$ predicted from decoder layers. Let the hand poses, object poses and probability of object types predicted by the i-th query in the l-th decoder layer as $\mathbf{h}_i^l$, $\mathbf{o}_i^l$,

$\mathbf{c}_i^l$, respectively. Then, the reference point of the i-th query in the $l+1$-th decoder layer is calculated as follows:

$$\mathbf{r}_i^{(l+1)} = \begin{cases} c^{\mathrm{H}}(\mathbf{h}_i^l), & \text{if } \mathbf{c}_i^l \text{ denotes hand,} \\ c^{\mathrm{O}}(\mathbf{o}_i^l), & \text{if } \mathbf{c}_i^l \text{ denotes object} \end{cases} \quad (5)$$

where $c^{\mathrm{H}}(\cdot)$, $c^{\mathrm{O}}(\cdot)$ are the operations to find the center of hand pose and the center of the object, respectively.

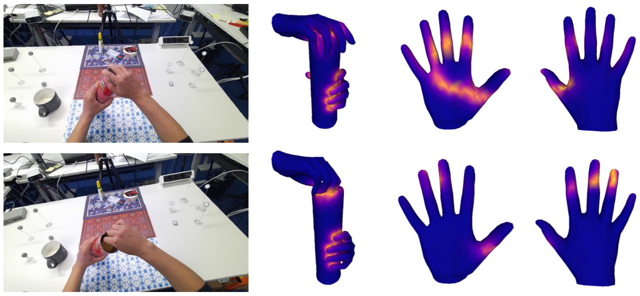
3.2. Hand-object interaction recognition

The hand-object interaction recognition network $f^{\mathrm{IA}}:[H,O,C]\rightarrow A$ is involved at this stage, to recognize the hand-object interaction classes $\mathbf{a}\in A$. We input the sequences of estimated hand poses $\mathbf{h}\in H$, object poses $\mathbf{o}\in O$ and object types $\mathbf{c}\in C$ combined with the contact map $\mathbf{m}\in M$ information as input to predict the interaction class $\mathbf{a}\in A$. First, we generate the hand and object vertices $\mathbf{V}=\{\mathbf{V}^{\mathrm{Left}},\mathbf{V}^{\mathrm{Right}},\mathbf{V}^{\mathrm{O}}\}$ and then generate the contact maps $\mathbf{m}=\{\mathbf{m}^{\mathrm{Left}},\mathbf{m}^{\mathrm{Right}},\mathbf{m}^{\mathrm{O}}\}$. Then, we map them towards the hand-object interaction class $\mathbf{a}$ via Transformer composed of encoder layers having the self-attention mechanism followed by the feedforward network.

3.2.1 Contact map generation

Benefits of contact maps. Variations in poses of hands and objects are important cues for recognizing interactions between hands and objects. However, we observed that recognizing interactions solely based on the poses of hands and objects is non-trivial in many cases. As shown in Fig. 3 (a), the first row denotes the frame from the video having the *open chips* interaction and the second row denotes the frame from the video having the *take out chips* in H2O dataset [32]. In both actions, hands move away from the object (i.e. chips) over time. Since the relative movement of hands and objects are similar in two videos, it is non-trivial to discriminate their interactions based solely on the pose information. On the contrary, in Fig. 3(b), we visualized the contact map that is able to explicitly represent the part where two hands and an object are contacted. We think this is a more effective clue to recognize the interaction between hands and objects even when their poses are similar.

Construction of contact maps. The contact map can be obtained using the meshes of two hands and an object. While the mesh estimation could be performed similarly to [4,23], we take the inverse kinematic (IK) approach which is more efficient. We propose to apply the HybrIK [35] method developed for the human body mesh reconstruction task in the hand domain to obtain the finger angle by finding the relative rotation matrix that rotates the template hand poses $\mathbf{t}=\{\mathbf{t}_k\}_{k=1}^K$ to the estimated hand poses $\mathbf{h}=\{\mathbf{h}_k\}_{k=1}^K$. Differently from the human bodies, each finger of hands do not involve the twist angles as fingers cannot be twisted respect to the bones; while it involves only swing angles. Exploiting the characteristics, we obtain the angles of the fingers from the estimated 3D poses $\mathbf{h}$ in the deterministic way, as follows:



(a) Video frame (b) Contact map for palm/back of a right hand

Figure 3. Example cases showing the importance of the contact map $\mathbf{m}$: The first row is a frame from the *open chips* video and the second row is a frame from the *take out chips* video in H2O dataset. The third and fourth columns show the contact map obtained from palm and back of the right hand. While poses are similar in two videos; the contact map activation could be a robust feature for discriminating interactions between hands and objects.

$$\mathbf{n}_k=\frac{\mathbf{t}_k\times\mathbf{h}_k}{\|\mathbf{t}_k\times\mathbf{h}_k\|},\cos\alpha_k=\frac{\mathbf{t}_k\cdot\mathbf{h}_k}{\|\mathbf{t}_k\|\cdot\|\mathbf{h}_k\|},\sin\alpha_k=\frac{\mathbf{t}_k\times\mathbf{h}_k}{\|\mathbf{t}_k\|\cdot\|\mathbf{h}_k\|}. \quad (6)$$

By the Rodrigues formula, the swing rotation matrix $\mathbf{R}_k$ is further derived as follows:

$$\mathbf{R}_k=\mathbf{I}+\sin\alpha_k[\mathbf{n}_k]_\times+(1-\cos\alpha_k)[\mathbf{n}_k]_\times^2 \quad (7)$$

where $\mathbf{I}$ is the 3×3 identity matrix and $[\mathbf{n}_k]_\times$ is the skew-symmetric matrix of $\mathbf{n}_k$. The rotation matrix $\mathbf{R}_k$ is converted into the 6D rotation representation and the hand pose parameters θ is obtained for MANO hand model [46].

We denote the vertices of generated left and right hand meshes as $\mathbf{V}^{\mathrm{Left}}\in\mathbb{R}^{778\times3}$,$\mathbf{V}^{\mathrm{Right}}\in\mathbb{R}^{778\times3}$, respectively. In addition, we sample $2{,}000$ vertex indices from the ground-truth object mesh and generate a sampled object vertices $\mathbf{V}^{\mathrm{O}}\in\mathbb{R}^{2{,}000\times3}$. Then, we transform them from the object space to the camera space using the predicted 6D poses. Finally, we use the distance-based encoding proposed in [27] to generate the contact map $\mathbf{m}$ that is composed of contact maps for left hand $\mathbf{m}^{\mathrm{Left}}\in\mathbb{R}^{778\times1}$, right hand $\mathbf{m}^{\mathrm{Right}}\in\mathbb{R}^{778\times1}$ and object $\mathbf{m}^{\mathrm{O}}\in\mathbb{R}^{2{,}000\times1}$. The formula that is used to calculate the i-th vertex of contact map $\mathbf{m}=\{\mathbf{m}^{\mathrm{Left}},\mathbf{m}^{\mathrm{Right}},\mathbf{m}^{\mathrm{O}}\}$, i.e. $\mathbf{m}_i$ is as follows:

$$\mathbf{m}_i=1-2\cdot(\rho(2D(\mathbf{V}_i,\mathbf{V}_j))-0.5) \quad (8)$$

where ρ is the sigmoid function, D is the distance function for calculating the distance between i-th vertex $\mathbf{V}_i$ and its nearest neighbor $\mathbf{V}_j$. When $\mathbf{V}_i$ denotes the vertex of hand contact maps (i.e. $\mathbf{m}^{\mathrm{Left}}$ or $\mathbf{m}^{\mathrm{Right}}$), the object vertex is used for $\mathbf{V}_j$, and when the $\mathbf{V}_i$ denotes the vertex of the object contact map (i.e. $\mathbf{m}^{\mathrm{O}}$), hand vertices are used for $\mathbf{V}_j$.

3.3. Mapping towards the actions

The input to the Transformer is the vector $\mathbf{v}_t$ that concatenates information at the t-th frame as follows:

$$\mathbf{v}_t \quad = \quad [\mathbf{v}_t^{\text{Left}\top};\mathbf{v}_t^{\text{Right}\top};\mathbf{v}_t^{\text{O}\top};\mathbf{o}_t^{\top}]^{\top} \tag{9}$$

where $\mathbf{o}_t$ denote the estimated object type probability at time t and $\mathbf{v}_t^{\text{Left}},\mathbf{v}_t^{\text{Right}},\mathbf{v}_t^{\text{O}} \in \mathbb{R}^{100\times 1}$ are obtained by projecting the vector that concatenates the mesh vertices $\mathbf{V}$ and contact maps $\mathbf{m}$ of left hand, right hand, and object in 100 dimensional space through the FC layers $f_{\text{L}},f_{\text{R}},f_{\text{O}}$ as follows:

$$\begin{aligned}\mathbf{v}_t^{\text{left}} &= f_{\text{L}}([\mathbf{V}_t^{\text{Left}};\mathbf{m}_t^{\text{Left}}]),\\ \mathbf{v}_t^{\text{right}} &= f_{\text{R}}([\mathbf{V}_t^{\text{Right}};\mathbf{m}_t^{\text{Right}}]),\\ \mathbf{v}_t^{\text{O}} &= f_{\text{O}}([\mathbf{V}_t^{\text{O}};\mathbf{m}_t^{\text{O}}]).\end{aligned} \tag{10}$$

We use the learnable action token $\alpha \in \mathbb{R}^{(300+N_c)\times 1}$ and vector $\mathbf{v}_t$ at each time t as inputs of the Transformer layer. The action token is refined by aggregating information about all frames through the self-attention. The action token of the last layer predicts the interaction class $\mathbf{a} \in A$ through the MLP. Since f^{IA} can model the long-range contextual relationship in a video, it is possible to recognize the interaction classes by capturing changes in hand-object interactions over time.

3.4. Training H2OTR

Since the Transformer-based methods have the high computational cost, we separately train the hand-object pose estimator f^{HOP} and the hand-object interaction classifier f^{IA} in two stages.

Training hand-object pose estimator f^{HOP}. Our hand-object pose estimation network f^{HOP} predicts N prediction sets of $\{\mathbf{y}_i\}_{i=1}^{N}$ where $\mathbf{y}_i = \{\mathbf{h}_i,\mathbf{o}_i,\mathbf{c}_i\}$. Let $\mathbf{y}^{\text{GT}} = \{\mathbf{y}_k^{\text{GT}}\}_{k=1}^{K}$ is the set of $\mathbf{y}_k^{\text{GT}} = \{\mathbf{h}_k^{\text{GT}},\mathbf{o}_k^{\text{GT}},\mathbf{c}_k^{\text{GT}}\}$ which is the ground-truth set involving K instances. Depending on whether the instance is a hand or an object, either $\mathbf{h}^{\text{GT}}$ or $\mathbf{o}^{\text{GT}}$ is exploited while setting unused ones as the zero vector. Since N is set as larger number than the normal number of instances K in the image, we pad the zero vector $\emptyset$ with $\mathbf{y}^{\text{GT}}$ to make it to the N sets. We search for the permutation of the N elements $\sigma \in \mathfrak{S}_N$ with the lowest cost to find a bipartite matching between these two sets.

$$\hat{\sigma} \quad = \quad \underset{\sigma\in\mathfrak{S}_N}{\operatorname{argmin}} \sum_{i}^{N} C_{\text{match}}, \tag{11}$$

where the matching cost C_{match} is defined measuring the distance between the ground truth $\mathbf{y}_k$ and the prediction $\mathbf{y}_{\sigma(k)}$ as follows:

$$\begin{aligned}C_{\text{match}} \quad &= \quad \mathbb{1}_{\{\mathbf{c}_k\neq 0\}}\cdot -\log([f_{\text{hand}}(\mathbf{z}'_{\sigma(k)})]_{\mathbf{c}_k})\\ &+ \quad \mathbb{1}_{\{\mathbf{c}_k\in \mathbf{H}\}}\cdot \|f_{\text{hand}}(\mathbf{z}'_{\sigma(k)}) - \mathbf{h}_k^{\text{GT}}\|_1\\ &+ \quad \mathbb{1}_{\{\mathbf{c}_k\in \mathbf{O}\}}\cdot \|f_{\text{obj}}(\mathbf{z}'_{\sigma(k)}) - \mathbf{o}_k^{\text{GT}}\|_1\end{aligned} \tag{12}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function that outputs 1 only when the statement is true, output 0 otherwise. The cost consists of classification score, hand and object pose scores. Also, it is calculated only when the label $\mathbf{c}_k$ is not 0 indicating the background.

For a permutation $\hat{\sigma}$ where the ground-truth set $\mathbf{y}^{\text{GT}}$ is best matched to, we applied the hungarian loss L_{H} to train the hand-object pose estimation network f^{HOP} as follows:

$$\begin{aligned}L_{\text{H}}(f^{\text{HPO}}) \quad &= \quad \mathbb{1}_{\{\mathbf{c}_k\neq 0\}}\cdot -\log([f_{\text{hand}}(\mathbf{z}'_{\hat{\sigma}(k)})]_{\mathbf{c}_k})\\ &+ \quad \mathbb{1}_{\{\mathbf{c}_k\in \mathbf{H}\}}\cdot \|f_{\text{hand}}(\mathbf{z}'_{\hat{\sigma}(k)}) - \mathbf{h}_k^{\text{GT}}\|_1\\ &+ \quad \mathbb{1}_{\{\mathbf{c}_k\in \mathbf{O}\}}\cdot \|f_{\text{obj}}(\mathbf{z}'_{\hat{\sigma}(k)}) - \mathbf{o}_k^{\text{GT}}\|_1.\end{aligned} \tag{13}$$

The hungarian loss is calculated for all pairs $(k,\hat{\sigma}(k))$ in the best permutation $\hat{\sigma}$.

Training hand-object interaction recognizer f^{IA}. The hand-object interaction recognizer f^{IA} predicts the interaction class probabilities $\mathbf{a}$. To train it, we used the cross-entropy loss L_{action} that defines the distance between the ground truth action $\mathbf{a}^{\text{GT}} = \{\mathbf{a}_i^{\text{GT}}\}_{i=1}^{N_a}$ and the predicted interaction class probability $\mathbf{a} = \{\mathbf{a}_i\}_{i=1}^{N_a}$ as follows:

$$L_{\text{action}}(f^{\text{IA}}) = -\sum_{i=1}^{N_a} \mathbf{a}_i^{\text{GT}} \cdot \log(\mathbf{a}_i). \tag{14}$$

4. Experiments

We use the Pytorch for our implementation. The size of the input image is set to be 960×540 and the random rotation is applied as a data augmentation. We use AdamW optimizer with a different learning rate for each network: the learning rate of 2×10^{-4} is used for Transformer, 2×10^{-5} is used for the backbone network and weight decay of 1×10^{-4} is used. We used 4 RTX 3090 GPUs and set the batch size as 8 for each GPU.

4.1. Datasets and evaluation metrics

We evaluated our method on two datasets: H2O [32] and FPHA [18]. Both datasets provide 3D hand poses, object 6D poses, object types and interaction classes.

H2O. The H2O dataset involves an interacting scenario with two hands and an object, and it contains 4 subjects and 8 objects. This provides a multi-view images including an egocentric view, but we only used images from the egocentric view. For our pose estimation step, We used 55,742 images to train, 11,638 to validate and 23,391 to test. For interaction recognition, we used 569 video clips to train, 122 to validate, 242 to test. The test split contains only 1 subject which is unseen during training.

FPHA. The FPHA dataset provides annotations for only one hand and an object in the egocentric view. We use the action split by 1:1 ratio, and the trainset consists of 600 videos and the testset consists of 575 videos for pose estimation. The FPHA provides 3D annotations for only 4 objects involving 10 interaction classes. Since our method needs the object mesh, we use the subset where object 3D annotation is available for interaction recognition.

Evaluation metrics. For hand and object pose estimation, we compute the mean end-point error (mm) over 21 poses. The error measures the Euclidean distance between predictions and

Table 1. Quantitative comparison to state-of-the-art methods for pose estimation on test sets of H2O and FPHA datasets. Since [23, 50] are single hand methods, they reported results separately for left and right hand-objects. Our method outperforms previous methods with a significant margin. Best results are bold-faced.

Method	H2O			FPHA	
	Left.h	Right.h	Object L/R	Right.h	Object
Hasson et al. [22]	39.6	41.9	67.5/66.1	18.0	22.3
Tekin et al. [50]	41.4	38.9	48.1/52.6	15.8	24.9
Kwon et al. [32]	41.5	37.2	47.9	-	-
Wen et al. [56]	35.0	36.1	-	15.8	-
Aboukhadra1 et al. [1]	36.8	36.5	73.9	-	-
Ours	**24.4**	**25.8**	**45.2**	**15.0**	**21.0**

Table 2. Quantitative comparison to state-of-the-art methods for interaction recognition on test sets of H2O and FPHA. Best results are bold-faced.

Method	H2O	FPHA
	Acc.	Acc.
C2D [54]	70.7	-
I3D [8]	75.2	-
SlowFast [17]	77.7	-
Tekin et al. [50]	68.9	97.0
Kwon et al. [32]	79.3	-
Wen et al. [56]	86.4	-
Ours	**90.9**	**98.4**

Table 3. Ablation study on the reference point refinement for the hand-object pose estimator f^{HOP}.

-		w/o refinement	w/ refinement
H2O	Left.h	27.9	**24.4**
	Right.h	31.1	**25.8**
	Object	51.0	**45.2**
FPHA	Right.h	24.8	**15.0**
	Object	40.8	**21.0**

Table 4. Ablation study on hand-object interaction recognizer f^{IA}.

Modality	Sampling points	H2O		FPHA
		Val Acc.	Test Acc.	Test Acc.
Pose	-	90.2	89.3	91.9
Mesh w/o Contact	2000	91.8	86.8	96.8
Mesh w/ Contact	500	87.7	85.1	96.8
	1000	91.0	87.6	96.8
	1500	91.8	88.4	**98.4**
	2000	**92.6**	**90.9**	**98.4**
Mesh w/Contact (10-NN)	2000	90.2	82.2	**98.4**

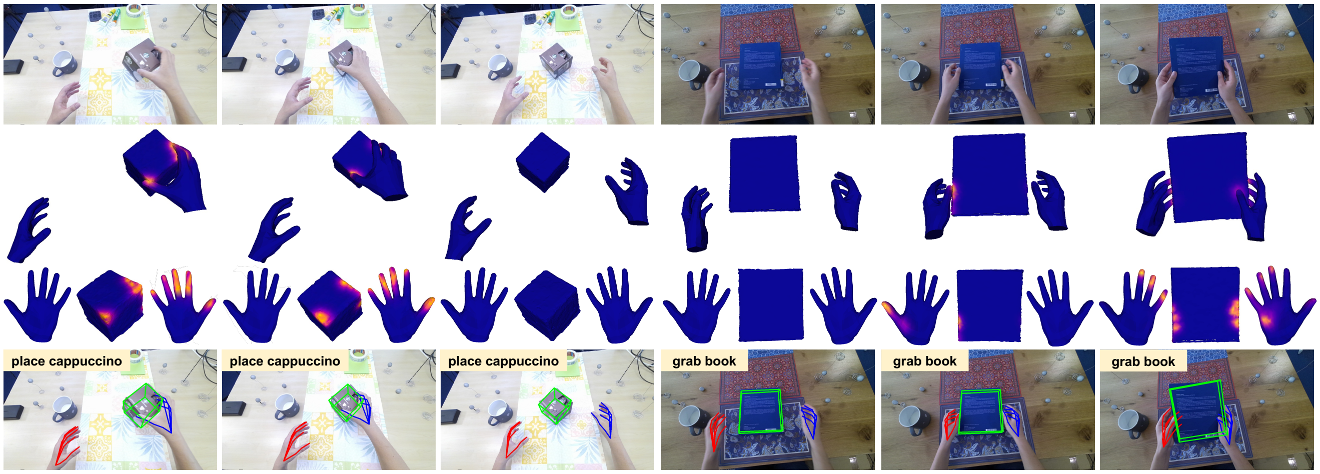


Figure 4. Examples of hand-object poses and interacting classes on H2O dataset predicted by our H2OTR. (Row 1) Input frame, (Row 2) Contact map in interaction space, (Row 3) Contact map in canonical poses, (Row 4) Estimated hand/object poses and interacting classes.

ground truths. For interaction recognition, we compute the top-1 accuracy. The top-1 accuracy is the conventional accuracy i.e., model prediction must be exactly the expected ground-truth.

4.2. Experimental results

Comparison with state-of-the-art methods for pose estimation. We compared our model with state-of-the-art methods which estimate the poses from full images on test sets of H2O and FPHA datasets. Table 1 summarizes the results for hand and object pose error. Hasson et al. [22] and Tekin et al. [50] predict single hand's pose and object pose. Also, Wen et al. [56] predicts only hand poses without object poses. Others predict for two hands and an object. Our method outperformed all previous works and achieved the state-of-the-art accuracy for hand-object pose estimation. Fig. 4 and 5 show the qualitative results of our framework on test sets in H2O and FPHA datasets, respectively. More qualitative results are accompanied in the supplemental pages and video.

Comparison with state-of-the-art methods for action recogniton. We compared the hand-object interaction recognition

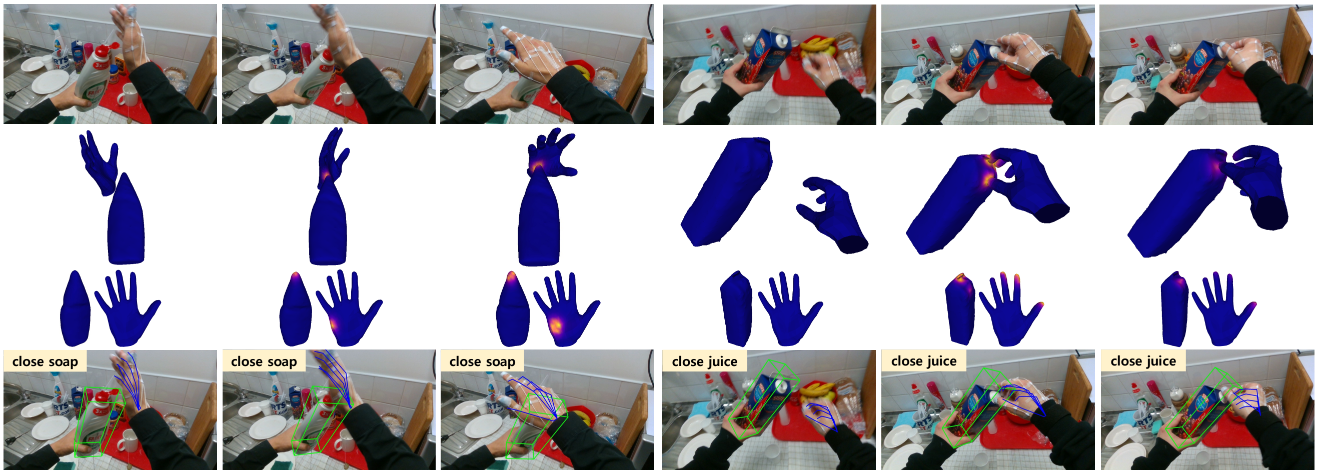


Figure 5. Examples of hand-object poses and interacting classes on FPHA dataset predicted by our H2OTR. (Row 1) Input frame, (Row 2) Contact map in interaction space, (Row 3) Contact map in canonical poses, (Row 4) Estimated hand/object poses and interacting classes.

accuracy with state-of-the-art methods summarized in Table 2. In [17, 54], they used only the RGB image sequences, and [8] used optical flow as additional information. [32, 50, 56] use the recurrent model, graph neural network, Transformer, respectively to recognize interaction class. In all works, the output from the pose estimator is fed into the interaction classifier. Since the dynamics of hand and object poses contain considerable information, it shows higher accuracy when using poses as modality. We shows better performance than the existing methods exploiting both predicted poses and contact maps.

Ablation studies. We verified our design choices in this subsection. Table 3 shows the performance difference depending on whether the reference point refinement (described in Sec. 3.1) is applied or not. When the refinment is applied, the predicted pose error is decreased a lot by 3.5mm in the left hand, 5.3mm in the right hand, and 5.8mm in the object. Table 4 shows the performance difference according to the modality used for the interaction recognition. We achieved the best performance when using both contact maps ('Contact') and mesh vertices ('Mesh'). 'Pose' denotes the raw 21 hand poses estimated. Additionally, we conducted experiments using the number of nearest vertices. Since using the relative distance between all vertices is infeasible to memory, we report the result of the experiment with 10 nearest vertices denoted as '10-NN'. While it is feasible in the memory, the accuracy (ie. 82.2%) is limited. Furthermore, we present the performance obtained by utilizing alternative architectures, such as CNN for pose estimation and LSTM for interaction recognition, instead of a transformer-based approach in Table 5. The 'CNN+f^{IA}' denotes a method combining CNN-based pose estimator [32] with our Transformer-based interaction recognizer f^{IA} and 'f^{HOP}+LSTM' denotes a method that combines our Transformer-based pose estimator f^{HOP} with the LSTM [25], respectively. This demonstrates the effectiveness of our pipeline that constitutes both parts as the Transformer-based architecture rather than the CNN or LSTM.

Table 5. Ablation study on architecture design.

Method	Left.h	Right.h	Object	Acc.
CNN [32] + f^{IA}	45.9	41.2	57.0	80.9
f^{HOP} + LSTM [25]	24.4	25.8	45.2	82.6
f^{HOP}+f^{IA}	**24.4**	**25.8**	**45.2**	**90.9**

5. Conclusion

In this paper, we propose a unified framework which consist of a hand-object pose estimator and a hand-object interaction recognizer. Our network performs 4 tasks simultaneously: hand and object pose estimation, object type classification, hand-object interaction recognition. We additionally proposed to use the contact map as a cue for hand-object interaction recognition. We achieved the state-of-the-art accuracy in every tasks and also demonstrated that each component works in the meaningful way.

6. Acknowledgements

This work was supported by IITP grants (No. 2021-0-01778 Development of human image synthesis and discrimination technology below the perceptual threshold 20%; No. 2020-0-01336 Artificial intelligence graduate school program (UNIST) 10%; No. 2021-0-02068 Artificial intelligence innovation hub 10%; No. 2022-0-00264 Comprehensive video understanding and generation with knowledge-based deep logic neural network 20%) and the NRF grant (No. 2022R1F1A1074828 10%), all funded by the Korean government (MSIT). This work was also supported by LG Electronics (30%).

References

[1] Ahmed Tawfik Aboukhadra, Jameel Malik, Ahmed Elhayek, Nadia Robertini, and Didier Stricker. Thor-net: End-to-end graformer-based realistic two hands and object reconstruction with self-supervision. arXiv:2210.13853, 2022. 7

[2] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, Neo Chen, Boshen Zhang, Fu Xiong, Yang Xiao, Zhiguo Cao, Junsong Yuan, Pengfei Ren, Weiting Huang, haifeng sun, Marek Hrúz, Jakub Kanis, Zdeněk Krňoul, Qingfu Wan, Shile Li, Dongheui Lee, Linlin Yang, Angela Yao, Yun-Hui Liu, Adrian Spurr, Pavlo Molchanov, Umar Iqbal, Philippe Weinzaepfel, Romain Brégier, Grégory Rogez, Vincent Lepetit, and Tae-Kyun Kim. Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction. In ECCV, 2020. 2

[3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In CVPR, 2018. 1, 2

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In CVPR, 2019. 1, 2, 5

[5] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-supervised domain adaptation via gan and mesh model for estimating 3d hand poses interacting objects. In CVPR, 2020. 1, 2

[6] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In ICCV, 2015. 1, 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020. 2, 3, 4

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, 2017. 2, 7, 8

[9] 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 CVPR, 2021. 1, 2

[10] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In CVPR, 2020. 1, 2

[11] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In WACV, 2021. 1, 2

[12] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In CVPR, 2022. 1, 2

[13] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In CVPR, 2020. 1, 2

[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. In ICLR, 2021. 3

[15] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In ICCV, 2011. 2

[16] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In CVPR, 2011. 2

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In ICCV, 2019. 2, 7, 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 CVPR, 2018. 1, 2, 6

[19] John C Gower. Generalized procrustes analysis. Psychometrika, 1975. 3

[20] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In CVPR, 2020. 1, 2

[21] 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 CVPR, 2022. 3

[22] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In CVPR, 2020. 1, 7

[23] 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 CVPR, 2019. 1, 2, 5, 7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 3, 4

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 1997. 1, 8

[26] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Real-time iterative rendering and refinement for 6d object pose estimation. In ICCV, 2021. 1, 2

[27] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In CVPR, 2021. 2, 5

[28] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In ICCV, 2017. 1, 2

[29] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In CVPR, 2021. 3

[30] Bumsoo Kim, Jonghwan Mun, Kyoung-Woon On, Minchul Shin, Junhyun Lee, and Eun-Sol Kim. Mstr: Multi-scale transformer for end-to-end human-object interaction detection. In CVPR, 2022. 3

[31] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. End-to-end detection and pose estimation of two interacting hands. In ICCV, 2021. 1, 2

[32] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In CVPR, 2021. 1, 2, 5, 6, 7, 8

[33] Seongyeong Lee, Hansoo Park, Dong Uk Kim, Jihyeon Kim, Muhammadjon Boboev, and Seungryul Baek. Image-free domain generalization via clip for 3d hand pose estimation. In WACV, 2023. 1

[34] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In CVPR, 2022. 3

[35] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In CVPR, 2021. 5

[36] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In CVPR, 2019. 1, 2

[37] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In CVPR, 2021. 3

[38] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In ICCV, 2021. 1, 2, 3

[39] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In CVPR, 2021. 1, 2

[40] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In ICLR, 2022. 3

[41] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In ECCV, 2022. 3

[42] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In CVPR, 2016. 1, 2

[43] 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 ECCV, 2020. 1

[44] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In CVPR, 2012. 2

[45] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In CVPR, 2017. 1, 2

[46] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In SIGGRAPH Asia, 2017. 5

[47] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In CVPR, 2019. 2

[48] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In CVPR, 2016. 1, 2

[49] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In CVPR, 2021. 3

[50] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In CVPR, 2019. 1, 2, 7, 8

[51] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In CVPR, 2018. 1, 2

[52] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015. 2

[53] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video. In SIGGRAPH, 2020. 1, 2

[54] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In CVPR, 2018. 7, 8

[55] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In AAAI, 2022. 3

[56] Yilin Wen, Hao Pan, Lei Yang, Jia Pan, Taku Komura, and Wenping Wang. Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos. arXiv:2209.09484, 2022. 7, 8

[57] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv:1711.00199, 2017. 1, 2

[58] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In CVPR, 2022. 1, 2

[59] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. In ICLR, 2022. 3

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In ICLR, 2021. 2, 3, 4

[61] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In ICCV, 2017. 1, 2

[62] Shihao Zou, Yuanlu Xu, Chao Li, Lingni Ma, Li Cheng, and Minh Vo. Snipper: A spatiotemporal transformer for simultaneous multi-person 3d pose estimation tracking and forecasting on a video snippet. arXiv:2207.04320, 2022. 3