THOR-Net: End-to-end Graformer-based Realistic Two Hands and Object Reconstruction with Self-supervision

Ahmed Tawfik Aboukhadra[1,2] Jameel Malik[1,3] Ahmed Elhayek[4]
Nadia Robertini[1] Didier Stricker[1,2]

[1]DFKI-AV Kaiserslautern [2]TU Kaiserslautern [3]NUST-SEECS Pakistan [4]UPM Saudi Arabia

Abstract

Realistic reconstruction of two hands interacting with objects is a new and challenging problem that is essential for building personalized Virtual and Augmented Reality environments. Graph Convolutional networks (GCNs) allow for the preservation of the topologies of hands poses and shapes by modeling them as a graph. In this work, we propose the THOR-Net which combines the power of GCNs, Transformer, and self-supervision to realistically reconstruct two hands and an object from a single RGB image. Our network comprises two stages; namely the features extraction stage and the reconstruction stage. In the features extraction stage, a Keypoint RCNN is used to extract 2D poses, features maps, heatmaps, and bounding boxes from a monocular RGB image. Thereafter, this 2D information is modeled as two graphs and passed to the two branches of the reconstruction stage. The shape reconstruction branch estimates meshes of two hands and an object using our novel coarse-to-fine GraFormer shape network. The 3D poses of the hands and objects are reconstructed by the other branch using a GraFormer network. Finally, a self-supervised photometric loss is used to directly regress the realistic textured of each vertex in the hands' meshes. Our approach achieves State-of-the-art results in Hand shape estimation on the HO-3D dataset (10.0mm) exceeding ArtiBoost (10.8mm). It also surpasses other methods in hand pose estimation on the challenging two hands and object (H2O) dataset by 5mm on the left-hand pose and 1 mm on the right-hand pose. THOR-Net code will be available at `https://github.com/ATAboukhadra/THOR-Net`.

1. Introduction

Realistic hands-object shape reconstruction is crucial for many Augmented Reality (AR) and Virtual Reality (VR) applications in order to create a more immersive, personalized experience for the users. Moreover, the hand pose is useful for human-computer interaction, action recognition, human behavior analysis, and gesture recognition applications [9, 28, 2, 6, 1, 26]. The recent advancements in hand, body and object pose estimation [17, 30, 14, 15, 28] are promising. However, few attention is given to the joint reconstruction of two hands interacting with an object [15, 17, 2, 9, 36]. This is a challenging problem due to varying hand shapes, texture, many degrees of freedom (DOF), self-similarity of hands parts, two-hands self-occlusions, and hand-object mutual occlusion, especially from a monocular RGB image as it only contains 2D information.

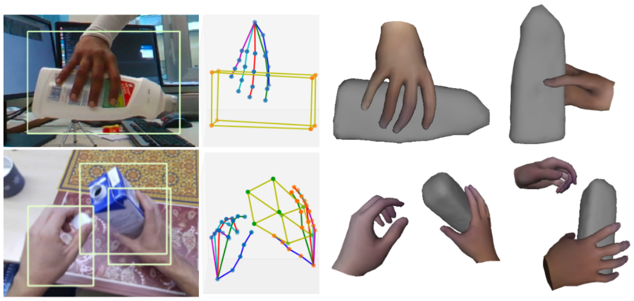

Figure 1. Our Graformer-based algorithm jointly reconstructs up to two hands Poses and textured shapes together with a shape of one object from a monocular RGB image. Note that the hands' textures of the above shapes were directly regressed for each vertex based on self-supervision training.

By utilizing the recent advances in deep learning (e.g., GCNs, Transformers, and self-supervised learning), several algorithms for simultaneous hand pose and shape estimation have been introduced. Recently, many researchers used Graph Convolutional Networks (GCNs) [39] to address the challenges of pose estimation [9, 41, 43, 4] and shape reconstruction [2, 28, 38]. GCNs preserve the inherent kinematic and graphical structure of hand pose and shape. This feature allows GCNs to handle depth ambiguity and occlusions as it correlates the visible parts of the hand with the non-visible parts [9]. Transformer networks [37] have also shown great abilities in many domains such as NLP [8]. Transformers have shown to be highly effective in many Computer Vision domains [10]. Many researchers have studied the effectiveness of Transformers in hand pose and shape estimation [20, 30, 43, 14, 40, 23].

In this paper, we propose the first —to the best of our knowledge— approach with GCNs, Transformers, and self-supervision which simultaneously estimates the 3D shape and the 3D pose of two hands interacting with an object together with the texture of each vertex of the hands given a monocular RGB image as shown in Figure 1.

THOR-Net is based on Keypoint RCNN which extracts several 2D features (i.e., heatmaps, bounding boxes, features maps, and 2D pose) from the monocular RGB image. To benefit from the power of the GCNs we model all this 2D information as two graphs. One graph is passed through our novel coarse-to-fine GraFormer shape generator network to estimate meshes for the hands and the object. This network gradually increases the number of nodes in the graph starting from the pose until reaching the shape while gradually decreasing the size of the features to only 3 values (x,y,z) that correspond to each vertex location in 3D space. The other graph is passed through a 2D-to-3D pose estimation network which is based on GraFormer to estimate 3D poses for the hands and object.

The hands' textures of the meshes are directly regressed by using a self-supervision photometric loss. To this end, the texture of each vertex is learned by orthographic projection to the input image. In contrast to HTML [31] which learns the statistical hand texture model from a limited set of hand texture samples, our photometric loss approach allows for learning hand textures from a huge set of RGB images of any hands dataset.

To summarize, we make the following contributions:

- A novel pipeline to reconstruct a realistic 3D shape for two hands and objects from RGB images with the following novelties:
  - Utilizing heatmaps and features produced by the Keypoint RCNN to build graphs that help our GraFormer-based networks to estimate 3D pose and shape.
  - Proposing a coarse-to-fine GraFormer for two hands and object reconstruction.
- Applying self-supervision based on photometric loss to give a more realistic view of hands.
- Our method achieves state-of-the-art results for hand mesh estimation on HO-3D (v3) and hand pose estimation on the H2O dataset as shown in Section 4.

2. Related Work

Although most of the existing works focus on the reconstruction of a single interacting hand, our work addresses a more challenging problem of two hands and object reconstruction. Here, we briefly describe the most related works.

2.1. GCNs for Pose Estimation

Recently, 3D pose estimation from 2D pose using Graph Convolutional Networks (GCNs) showed very promising results [9, 43]. Using a single Keypoint from the 2D pose to estimate its counterpart in 3D is a nondeterministic problem. However, using the information about other 2D keypoints and their relation to the target keypoint can be useful to estimate its 3D location. The authors of the HopeNet [9] introduced an adaptive GraphUNet that pools the 2D pose in five stages, and then unpools it to get the 3D pose while having skip connections between the corresponding pooling and unpooling layers.

The GraFormer [43] transforms 2D poses to 3D, however, it shows a much better performance than the HopeNet because of combining Graph Convolutional layers with the Transformer [37] and attention mechanism. The GraFormer is able to extract local features from the nodes using graph convolutional layers and also extract global information about the entire graph using the attention layers.

The spatiotemporal graph solves the depth ambiguity and severe occlusion challenges in 3D pose estimation [4, 41]. Temporal continuity in videos imposes temporal constraints [15]. Therefore, Cai *et al.* [4] created a Spatiotemporal graph from a few temporally adjacent 2D body poses by creating additional edges between the joints and their counterparts in neighboring frames.

2.2. Hand-Object Reconstruction

Most of the existing works focus on hand shape estimation under interaction with an object without considering object shape reconstruction. The Keypoint Transformer [14] achieves state-of-the-art results in hand pose estimation from RGB images by extracting features from the image for each keypoint and correlating those features using self-attention layers. HandOccNet [30] is a very recent and robust transformer-based model that solves the ambiguity of occlusions between hands and objects by injecting features from visible areas of the hand to areas where the hand is occluded by the object. ArtiBoost [42] aims to solve the lack of diversity of hand-object poses within the 3D space in any hand-object dataset by creating synthetic images. They use both synthetic and real images to train a CNN regression model that estimates the pose. Liu *et al.* [24] leveraged spatiotemporal consistency in RGB videos to generate labels for semi-supervised training to estimate 3D pose.

Two-hands and object reconstruction does not get enough attention compared to hand-object pose estimation

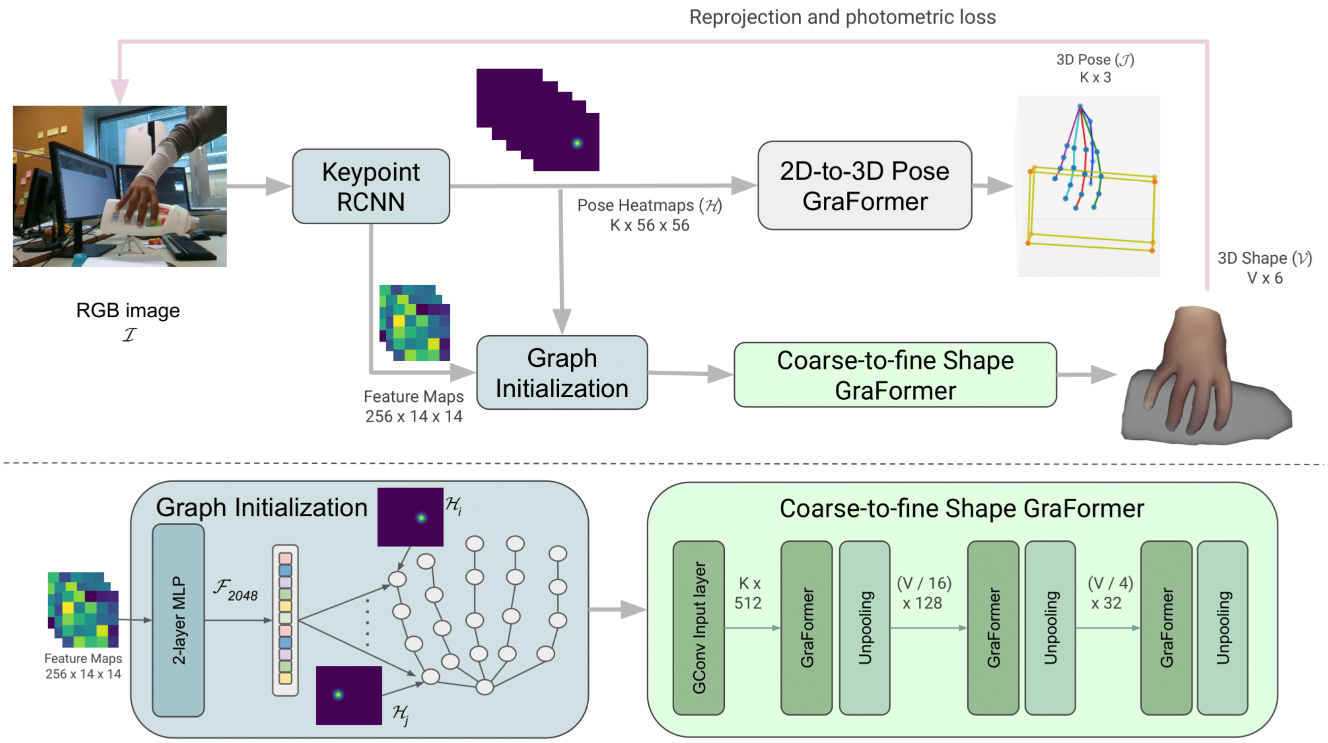

Figure 2. An overview of our approach to estimating the 3D pose and 3D shape for hands interacting with an object from a monocular RGB frame. K is the number of keypoints in the Pose and V is the number of vertices in the Shape. The lower part describes more details about the graph initialization and the Coarse-to-fine Shape GraFormer network.

and hands-only reconstruction. Hasson *et al.* [15] used a network that outputs the MANO [34] parameters for the hand and the object class with its 3D transformation parameters. One important aspect of their work is that they use photometric consistency over time as a semi-supervised training scheme when some frames are not annotated. In their follow-up work [16], Hasson *et al.* first detect and segment hands and objects within an RGB image. After that, they estimate hand shape and object pose and optimize them using loss terms for smoothness and collision.

Malik *et al.* [27, 25] investigated hand pose and shape estimation from depth maps. [25, 28] used voxelized depth maps to estimate a voxelized shape and a shape surface for the hand, followed by a registration step. EventHands [35] is a network that uses an Event camera input to capture and reconstruct hand motions of unprecedented speed. Almadani *et al.* [2] created a depth-based coarse-to-fine hand object reconstruction network that is built on the GCN HopeNet [9]. After evaluating different input modalities for their model, they show that a voxelized representation of a depth map and a corresponding RGB image is the best input modality. Pixel2Mesh [38] is a GCN network that estimates 3D shapes for objects from monocular RGB frames.

3. Method

The proposed pipeline is shown in Figure 2, it uses RGB frame $\mathcal{I}$ as an input and predicts the target 3D pose $\mathcal{J}$ and the 3D shape $\mathcal{V}$ for hands and objects.

3.1. Keypoint RCNN

Mask RCNN [18] is effective object detection and semantic segmentation model built over Faster RCNN [33]. Mask RCNN proposes Regions-of-interest (RoIs) within an image that contains objects and estimates the bounding box and the class for those objects. The authors of the Mask RCNN created a variant called Keypoint RCNN that estimates heatmaps of the location of any set of 2D keypoints within the RoI. For every keypoint there is a heatmap of the location of that keypoint. From bounding boxes and heatmaps, Keypoint RCNN can estimate 2D locations in the image that compose a 2D pose. We train the Keypoint RCNN to estimate the 2D pose for hands and objects knowing the projection of the 3D pose to 2D. Hence, the Keypoint RCNN provides important information from RGB images such as bounding boxes for hands and objects, heatmaps for keypoints within those boxes, and RoI features.

To train the Keypoint RCNN, bounding box annotations are required. To obtain the bounding boxes, we use the 2D

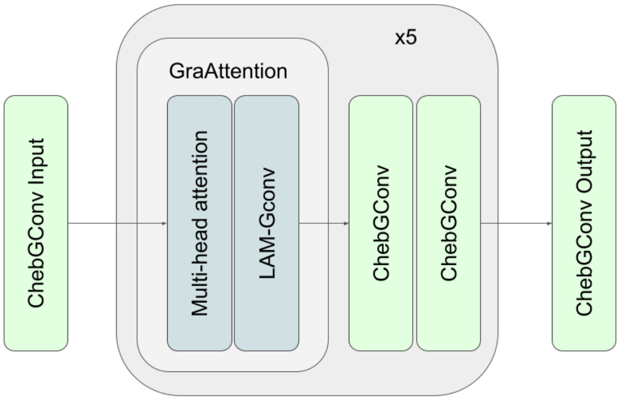

Figure 3. An illustration of the GraFormer network. The network consists of GCN and Attention layers repeated multiple times.

projection of the 3D pose to the image. The minimum x, and y values, and the maximum x, and y values of the 2D pose are considered to be the bounding box. Figure 2 shows the two outputs of the Keypoint RCNN which are the Heatmaps and the features, that are used to train our models for 3D pose and shape estimation. One important advantage of the Keypoint RCNN's ability to localize objects in raw images removes the need for image preprocessing such as cropping the hand-object region.

3.1.1 Feature Extractor

The backbone of the Keypoint RCNN consists of a ResNet50 [19] and a Feature Pyramid Network (FPN) [19] that produces multi-scale features for the RGB image. The backbone produces RoI-specific features after passing the multi-scale features into a multi-scale RoI align layer [18]. This allows us to capture custom features for the RoI that contains hands or objects. We use those features to enrich the nodes of the coarse-to-fine GraFormer. To compress the features before passing them to the next stage, the RCNN passes the features to a 2-layer MLP that produces a compressed 2048 feature vector for each RoI. These feature vectors are appended to the heatmaps to produce a graph representation for the shape generator, as described in Sec. 3.3.

3.2. Pose GraFormer

To convert the 2D information extracted by the Keypoint RCNN into 3D space, we use the GraFormer. The GraFormer [43] is a Graph Neural Network that is designed to utilize the advantages of Graph Convolutional layers and Attention layers [37]. Graph Convolutional layers extract features from graph-like data depending on the connectivity between the nodes, in our case, the edges between adjacent keypoints of the pose or the shape. Furthermore, the multi-head self-attention [37] layers within the GraFormer extract global features from the graph. Both concepts caused the GraFormer to outperform other methods in 2D-to-3D pose lifting.

As illustrated in Figure 3, the GraFormer consists of a GraAttention layer that is a multi-head self-attention layer with 4 heads. The last layer of the GraAttention is a LAM-GConv layer which is a graph convolution layer with a trainable adjacency matrix. The GraAttention is followed by 2 layers of a special type of graph convolutions called ChebGConv[7] composing the main building component of the GraForemr. This component of the GraAttention and the ChebGConv is repeated five times to create the GraFormer. The first use of the GraFormer in our work is to transform the heatmaps of the Keypoint RCNN to 3D pose coordinates of the hand and the object. Instead of representing every node in the graph using its 2D pixel location, we found empirically that using heatmaps is more accurate.

3.3. Coarse-to-fine Shape GraFormer

To generate the 3D shape, we propose a coarse-to-fine GraFormer that gradually increases the number of vertices starting from a 2D pose graph and ending with the 3D shape. Almadani *et al.* [2] and Wang *et al.* [38] previously explored Coarse-to-fine GCNs to generate 3D shapes. However, given the improved performance of GraFormer compared to normal GCNs, we replace their suggested Graph convolutional layers with GraFormers as shown in Fig. 2.

The network consists of three stages and each stage is composed of a GraFormer followed by an unpooling layer that increases the number of nodes in the graph. The input graph to the Coarse-to-fine GraFormer consists of 29 nodes. Every node *i* holds the feature vector $\mathcal{F}_{2048}$ and the corresponding heatmap $\mathcal{H}_i$ of size 56×56 as shown in Fig. 2. After flattening the heatmap and appending it with $\mathcal{F}_{2048}$, the size of the node representation is 5184.

To model a hand mesh as a graph, we use the MANO [34] faces to create the adjacency matrix. However, there are two challenges to creating such a coarse-of-fine graph network. The objects in both HO-3D and H2O datasets have a varying number of vertices and they do not have a consistent topology. The second challenge is that the intermediate graph layers within the coarse-to-fine network require a simplified version of the adjacency matrix as they have a lower number of vertices in their graphs. In Section 3.3.1, we describe how to create simplified versions of the hand mesh adjacency matrix. In section 3.3.2, we describe how to create a consistent topology for the objects.

3.3.1 Hand Mesh Downsampling

To create an intermediate graph representation of the hand, we use the Quadric Edge Collapse Decimation algorithm (QECD) [12, 29] to downsample the default MANO hand mesh. The faces of the resulting simplified mesh create the adjacency matrix of the intermediate graphs. We simplify

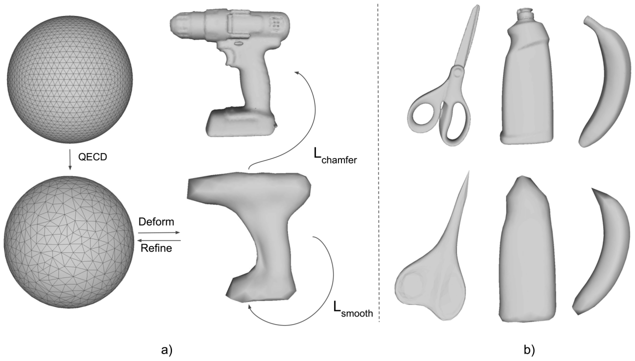

Figure 4. a) The process of simplifying and deforming a sphere to obtain a consistent topology representation for the mesh. b) Examples of different 3D object models and their simplified versions. The bottom row shows the downsampled spheres.

the 778 hand vertices to two granularity levels (i.e., 49 and 194) to correspond to levels 1 and 2 in the coarse-to-fine network respectively.

3.3.2 Object Topology

To solve the problem of the inconsistent object topology, we use a trainable approach from PyTorch3d [32] to deform a sphere with a constant topology to every object. An Icosphere is a sphere that results from the recursive subdivison of the polygons of a 20-faces polyhedron [29]. At level 4 of subdivision, the Icosphere has 2556 vertices. We use the QECD algorithm to simplify that sphere to 1000 vertices. This step is executed to control the number of vertices that represent the object shape depending on the complexity of the model and the required reconstruction quality.

To learn the displacement of every vertex in the sphere to the target object mesh, the deformation algorithm minimizes the Chamfer distance $\mathcal{L}_{chamfer}$ between the deformed sphere and the target mesh as shown in Figure 4. Along with the Chamfer loss, 3 additional regularization losses add smoothing effects on the resulting deformed sphere. The three losses are Edge length $\mathcal{L}_{edge}$, normal consistency of neighboring faces $\mathcal{L}_{norm}$ and Laplacian smoothing $\mathcal{L}_{laplacian}$. The final loss term for deformation is:

$$\mathcal{L} = \mathcal{L}_{chamfer} + \mathcal{L}_{edge} + \lambda_1 * \mathcal{L}_{norm} + \lambda_2 * \mathcal{L}_{laplacian} \quad (1)$$

λ_1 equals 0.01 and λ_2 equals 0.1. An SGD optimizer minimizes the weighted sum of the aforementioned losses until the sphere reaches the closest state to the target object. Figure 4 shows some of the objects in the YCB dataset [5], and their corresponding deformed sphere with 1000 vertices.

3.3.3 Photometric Loss

Estimating a texture value for every vertex in the mesh gives it a richer representation of a personalized hand shape and a more realistic view [31]. In addition, it helps to improve the alignment between the estimated shape with the target reprojection improving the reconstruction error. Furthermore, exploring hand textures is an interesting problem in the field of VR and AR as it improves the immersive experience. Qian *et al.* [31] proposed the first parametric hand texture model (HTML) for the reconstruction of realistic hand texture. Although this model allows the generation of a diverse set of hand textures by randomly sampling from the texture parameters, this approach is limited by the small training dataset (i.e., 51 subjects) which is used to build the hand texture model. This means that the proposed statistical model can not represent any texture that is not covered in this dataset. In this paper, we propose a direct texture regression approach that is based on self-supervision using photometric loss [15, 31]. To this end, a texture is directly learned for each hand mesh vertex together with the 3D position of this vertex. In contrast to HTML which learns the statistical hand texture model from a limited set of hand texture samples, our approach allows for learning hand textures from a huge set of RGB images of any hand dataset. From this motivation, we add an additional loss term to train the model, and instead of estimating only XYZ for the vertices, the model estimates an RGB value as well.

To calculate this loss, the target 3D shape is first projected into the image using the camera intrinsics. After that, the corresponding pixel RGB values of the projected vertices are penalized with the RGB values that the model estimates by calculating the MSE between both. This justifies the six values, as shown in Figure 2, for the final shape. The photometric loss $\mathcal{L}_{photo}$ is defined as follows:

$$\mathcal{L}_{photo} = MSE(\mathcal{I}[\mathit{proj}(\mathcal{V}_{gt})], \mathcal{V}_{pred,rgb}) \quad (2)$$

4. Experiments

In this section, we discuss the datasets and the implementation details for each dataset. After that, we discuss the training details and the loss functions. We then report and compare our results quantitatively and qualitatively with other methods in sections 4.4 and 4.5. Further, we conduct an ablation study to show the effectiveness of the components of our pipeline.

4.1. Datasets and Implementation Details

Researches have created many datasets recently to model the markerless hand-object interactions such as HO-3D [13], H2O [22], H2O-3D [14], DexYCB [6], FPHAB [11] and ContactPose [3]. We evaluate our method on two recent public benchmark datasets: HO-3D [13] and H2O [22]. HO-3D dataset has one hand interacting with an object while the H2O has two hands interacting with an object.

The HO-3D video dataset [13] contains 3D pose annotations for hand and a manipulated object under severe

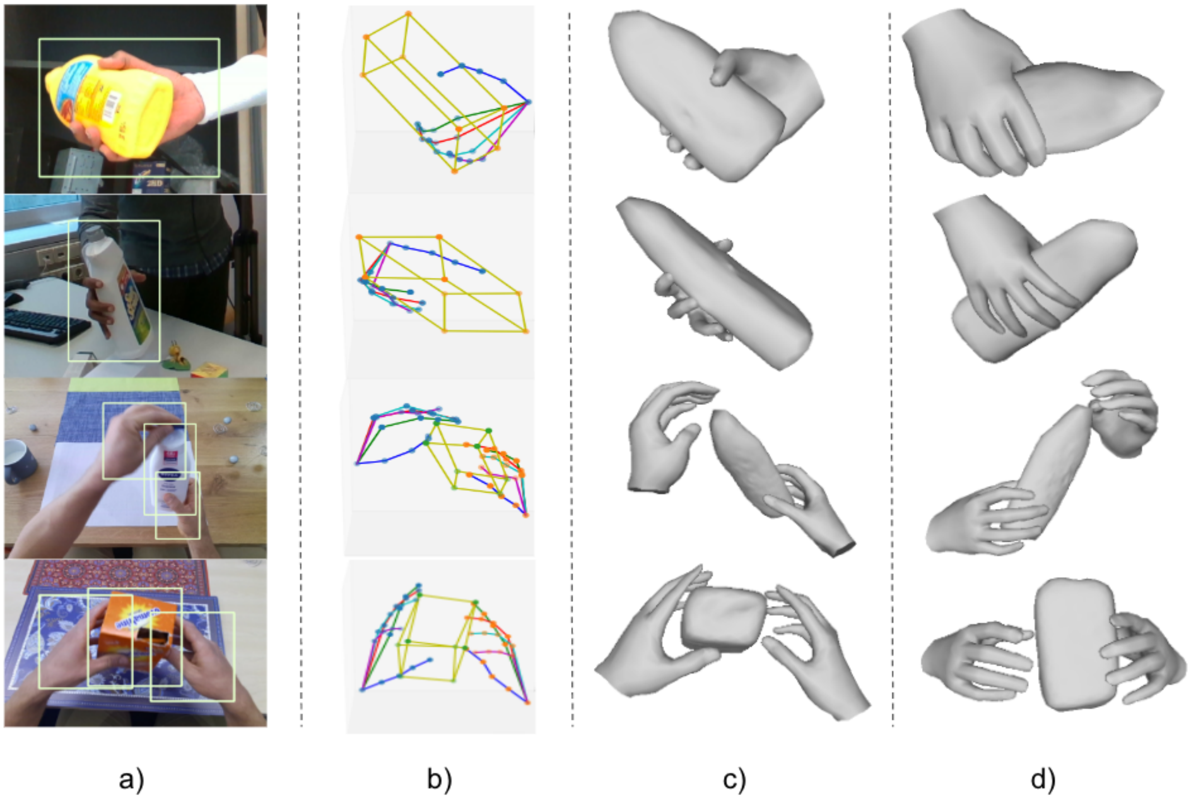

Figure 5. Qualitative results for Hand-Object pose and shape estimation from the HO-3D and H2O datasets. a) Input frame with the predicted bounding boxes. b) 3D pose estimation. c) & d) Two views for the 3D reconstructed hand-object interaction.

occlusions. The dataset also contains annotations for the MANO parameters and object labels. From the PyTroch Mano model[17], we acquire the hand vertices and faces and apply simplification to the hand to acquire the intermediate adjacency matrices for the coarse-to-fine network as described in section 3.3.1. All the 10 objects within the dataset were acquired from the YCB dataset [5]. We create a spherical representation for the objects following the description in 3.3.2, and use the pose to transform them into the 3D camera space. All the 3D points are translated such that the palm of the hand is used as the origin of 3D space. To train the Keypoint RCNN for HO-3D, we consider the hand and the object to be inside the same bounding box as shown in Figure 5. We report the results on the second and third versions of the dataset. The number of keypoints in the 3D pose $\mathcal{J}$ is 29; 21 for the hand joints and 8 for the object corners. The number of vertices in the 3D shape $\mathcal{V}$ is 1778; 778 for the hand mesh and 1000 for the object mesh.

H2O [22] is a new benchmark video dataset that contains the 3D pose annotations for two hands and an object along with the MANO parameters of the hands and the label of the object. The dataset covers 8 objects and provides a 3D model for all of them. We follow the same approach used with HO-3D to acquire the 3D mesh of the hands and the object in H2O. The dataset was captured from five different camera views, however, in this work, we only focus on the egocentric view.

To train the Keypoint RCNN for H2O, we separate each hand's bounding box from the object bounding box. We use the 2D keypoints produced by the Keypoint RCNN as an input to the GraFormer instead of the heatmaps as it has shown better results on the H2O dataset. The number of keypoints in the 3D pose $\mathcal{J}$ is 50; 21 for each hand's joints and 8 for the object corners. The number of vertices in the 3D shape $\mathcal{V}$ is 2556; 778 for each hand's mesh and 1000 for the object mesh.

4.2. Training

To train the THOR-Net, five losses are required: Cross Entropy Loss for heatmaps $\mathcal{L}_{\mathcal{H}}$, Bounding box classification $\mathcal{L}_{cls}$, Mean-squared Error (MSE) for bounding box estimation $\mathcal{L}_{bb}$, MSE to penalize the 3D pose $\mathcal{L}_{\mathcal{J}}$ and MSE to penalize the 3D shape $\mathcal{L}_{\mathcal{V}}$. We train our network with a combined loss function:

$$\mathcal{L} = \mathcal{L}_{\mathcal{H}} + \mathcal{L}_{cls} + \mathcal{L}_{bb} + \mathcal{L}_{\mathcal{J}} + \mathcal{L}_{\mathcal{V}} \tag{3}$$

In the case of generating textured shapes, we add the $\mathcal{L}_{photo}$ as described in Section 3.3.3. The network has 192M parameters and we train the model using an Adam

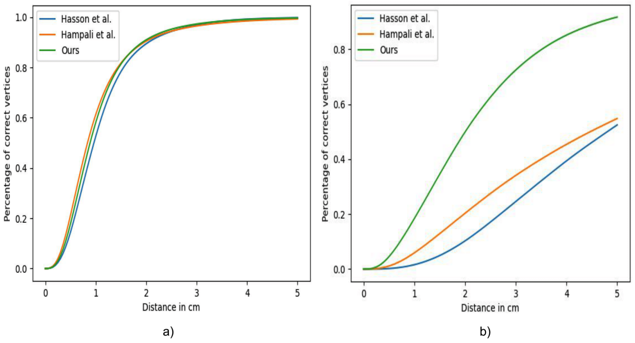

Figure 6. PCV over distance in comparison with other methods. a) Procrustes-aligned error. b) Non-aligned error.

optimizer [21], 0.0001 learning rate, and batch size 8 on an NVIDIA's A100 GPU.

4.3. Evaluation metrics

We report the Procrustes aligned and non-aligned MPJPE (Mean Per Joint Position Error) in *mm* on the HO-3D hand pose and shape and compare them with other methods in Tables 1 & 3. Furthermore, we show the percentage of correct vertices (PCV) over distance in Figure 6. To evaluate our model on the H2O dataset, we report the non-aligned MPJPE of the pose and shape for both hands and object in Table 2. Finally, we qualitatively show our results in Figure 5.

4.4. Evaluation of 3D Pose Estimation

We evaluate our method on the two versions of the HO-3D dataset and report the error in *mm* of the hand pose in Table 1. The table contains a comparison of the Procrustes aligned and non-aligned errors with the existing methods. The reported results can be found on the HO-3D challenge website [1]. We also evaluate our method for pose estimation on the H2O egocentric view and report the mean joint error for both hands and the object 3D pose in Table 2. The table shows the improvement in left and right-hand pose errors compared to previous methods. The results show an improvement of 5*mm* in left-hand pose estimation and 1*mm* in right-hand pose estimation compared to previous methods. The reported results can be found on the H2O challenge website [2]. Qualitative results of the 3D pose on samples from both datasets are shown in Figure 5. From the quantitative and qualitative evaluation, our pose estimation method requires future improvements as it does not exceed previous methods on HO-3D, and object pose estimation is not accurate on H2O.

Methods	$\mathcal{J}$ Al. Err.	$\mathcal{J}$ Err.
Hasson *et al.* [15]	11.4	55.2
Hasson *et al.* [17]	11.1	-
Hampali *et al.* [13]	10.7	84.2
METRO [23]	10.4	-
Liu *et al.* [24]	10.2	-
HandOccNet [30]	**9.1**	-
THOR-Net (Ours)	11.3	26.3
Keypoint Trans. [14]	10.9	-
ArtiBoost [42]	**10.8**	22.6
THOR-Net (Ours)	11.2	25.6

Table 1. **Comparison with state-of-the-art methods for 3D hand pose estimation on the HO-3D (v2) (upper table) and (v3) (lower table). Shown results are the Procrustes-aligned and non-aligned errors in *mm*.**

Methods	L $\mathcal{J}$ Err.	R $\mathcal{J}$ Err.	Obj. $\mathcal{J}$ Err.
Hasson *et al.* [15]	39.6	41.9	66.1
H+O [36]	41.4	38.9	48.1
H2O [22]	41.5	37.2	**47.9**
THOR-Net (Ours)	**36.8**	**36.5**	73.9

Table 2. **Comparison with the state-of-the-art methods for 3D pose estimation on H2O dataset. The shown results from left to right are the non-aligned errors in *mm* for the left-hand pose, the right-hand pose, and the object pose.**

4.5. Evaluation of 3D Shape Estimation

We evaluate our method for 3D shape estimation on two datasets and report the results in Table 3. The results show that our Procrustes-aligned mesh error is 10mm on the HO-3D (v3) while the best-found method achieves 10.8mm. To the best of our knowledge, we are the first to provide shape evaluations on the H2O dataset. The left-hand shape error is 54.1mm, the right-hand shape error is 59.4mm and the object shape error is 66.6mm.

Qualitative results of the hands-object shapes can be found in Figure 5. We also compare our hand shape results with Hasson *et al.* [15] in Figure 7. The results show that our model captures fine hand details. However, the model lacks the ability to generalize to unseen objects.

4.6. Ablation Study

We study the impact of the proposed coarse-to-fine GraFormer on the hand shape reconstruction by evaluating three versions of the shape generator network. As shown in Figure 2, the coarse-to-fine network consists of three GraFormers. To show the effectiveness of this choice, we test the network with 1 GraFormer and 2 GraFormers. Those two experiments are shown in Table 4 with IDs 1 and 2, respectively. The results show that the network with three GraFormers achieves the best performance. This shows that

[1] https://codalab.lisn.upsaclay.fr/competitions/4393

[2] https://codalab.lisn.upsaclay.fr/competitions/4822

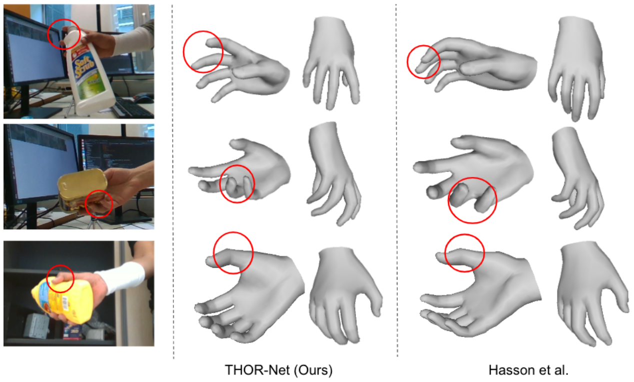

Figure 7. Qualitiative comparison with Hasson *et al.* [15]. Our method captures better hand details.

Methods	$\mathcal{V}$ Al. Err.	$\mathcal{V}$ Err.
HandVoxNet++ [28]	-	27.0
Hasson *et al.* [15]	11.4	55.2
METRO [23]	10.4	-
Hasson *et al.* [17]	11.0	-
Hampali *et al.* [13]	10.6	83.4
Liu *et al.* [24]	9.8	-
HandOccNet [30]	**8.8**	-
THOR-Net (Ours)	10.7	26.3
ArtiBoost [42]	10.4	-
THOR-Net (Ours)	**10.0**	23.7

Table 3. **Comparison with state-of-the-art methods for 3D hand shape estimation on the HO-3D (v2) (upper table) and (v3) (lower table). Shown results are the Procrustes-aligned and non-aligned errors in *mm*.**

the gradual increase of the deep coarse-to-fine network is useful for shape estimation.

To justify the choice of appending heatmaps to a feature vector of size 2048 as the initial Coarse-to-fine graph, we test four other different graph input modalities. Instead of using the heatmaps, we try to use either the estimated 2D pose or the 3D pose. In addition, we try two different feature vector sizes (i.e., 1024 and 4096) to test the capacity of the model and its correlation to the results. From the results shown in Table 4, it is clear that the heatmap representation along with the feature vector of size 2048 produces the best accuracy.

4.7. Quality of textured Shapes

Figure 8 shows the quality of the per-vertex texture values produced by the photometric loss. The network was able to reconstruct the hands despite the occlusions. Furthermore, it managed to capture some of the text details and the blue color of the book. Sometimes, a smoothing effect for the object colors happens as shown in the milk bottle hiding the details. In addition, the hands are affected by lighting conditions which cause different tones of skin color. The object's edges have artifacts as shown in the blue book.

ID	# GraFo.	Gr. Inp.	$\mathcal{V}$ Al. Err.	$\mathcal{V}$ Err.
1	1	$\mathcal{H} + \mathcal{F}_{2048}$	11.4	26.8
2	2	$\mathcal{H} + \mathcal{F}_{2048}$	11.1	26.4
4	3	2D pose + $\mathcal{F}_{2048}$	14.6	46.8
3	3	3D pose + $\mathcal{F}_{2048}$	10.9	27.0
5	3	$\mathcal{H} + \mathcal{F}_{1024}$	13.5	29.5
6	3	$\mathcal{H} + \mathcal{F}_{4096}$	11.8	28.4
7	3	$\mathcal{H} + \mathcal{F}_{2048}$	**10.0**	**23.7**

Table 4. **Ablation study for the depth of the Coarse-to-fine shape generator and the graph input modality.**

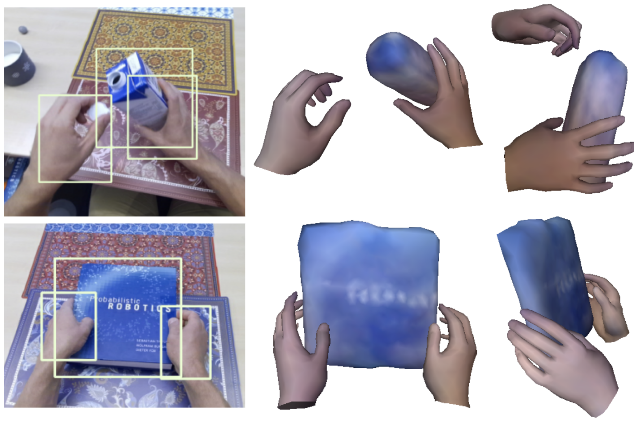

Figure 8. Examples of textured hands and objects estimated using photometric loss.

5. Conclusion and Future Work

In this work, we propose THOR-Net to reconstruct realistic two hands interacting with an object from a monocular RGB frame. The network consists of two stages: 2D feature extraction using Keypoint RCNN and 3D reconstruction using a coarse-to-fine GraFormer network. To obtain a per-vertex texture for shapes, we train the network using self-supervised photometric loss. The quantitative and qualitative evaluation shows the effectiveness of our coarse-to-fine network in two-hands-object shape estimation compared to previous methods. Qualitative results of photometric loss show a smoothing effect in texture estimation which suggests more future investigation. For future work, temporal constraints from videos can be leveraged to model a Spatio-temporal graph that can improve the reconstructions.

Acknowledgments This work has been partially funded by the German BMBF project GreifbAR (Grant Nr 16SV8732), and by the EU project FLUENTLY (Grant Nr 101058680).

References

[1] Mhd Rashed Al Koutayni, Vladimir Rybalkin, Jameel Malik, Ahmed Elhayek, Christian Weis, Gerd Reis, Norbert Wehn, and Didier Stricker. Real-time energy efficient hand pose estimation: A case study. *Sensors*, 20(10), 2020.

[2] Murad Almadani, Ahmed Elhayek, Jameel Malik, and Didier Stricker. Graph-based hand-object meshes and poses reconstruction with multi-modal input. *IEEE Access*, 9, 2021.

[3] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision (ECCV)*, 2020.

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2019.

[5] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics (ICAR)*, 2015.

[6] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[9] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[11] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.

[13] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] 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 *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Yana Hasson, Gül Varol, Cordelia Schmid, and Ivan Laptev. Towards unconstrained joint hand-object reconstruction from rgb videos. In *International Conference on 3D Vision (3DV)*, 2021.

[17] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[22] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *International Conference on Computer Vision (ICCV)*, 2021.

[23] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[26] Jameel Malik, Ahmed Elhayek, Sheraz Ahmed, Faisal Shafait, Muhammad Imran Malik, and Didier Stricker. 3dairsig: A framework for enabling in-air signatures using a multi-modal depth sensor. *Sensors*, 18(11), 2018.

[27] Jameel Malik, Ahmed Elhayek, and Didier Stricker. Whsp-net: A weakly-supervised approach for 3d hand shape and pose recovery from a single depth image. *Sensors*, 19(17), 2019.

[28] Jameel Malik, Soshi Shimada, Ahmed Elhayek, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker.

Handvoxnet++: 3d hand shape and pose estimation using voxel-based neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[29] Alessandro Muntoni and Paolo Cignoni. PyMeshLab, 2021.

[30] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[31] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization. In *European Conference on Computer Vision (ECCV)*, 2020.

[32] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. 2020.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[34] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[35] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. Eventhands: Real-time neural 3d hand pose estimation from an event stream. In *International Conference on Computer Vision (ICCV)*, 2021.

[36] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

[38] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *European Conference on Computer Vision (ECCV)*, 2018.

[39] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[40] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022.

[41] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[42] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. ArtiBoost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[43] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.