Joint Hand-Object Pose Estimation with Differentiably-Learned Physical Contact Point Analysis

Nan Zhuang, Yadong Mu*
{zhuangn53,myd}@pku.edu.cn
Wangxuan Institute of Computer Technology, Peking University, Beijing, China

ABSTRACT

Hand-object pose estimation aims to jointly estimate 3D poses of hands and the held objects. During the interaction between hands and objects, the position and motion of keypoints in hands and objects are tightly related and there naturally exist some physical restrictions, which is usually ignored by most previous methods. To address this issue, we propose a learnable physical affinity loss to regularize the joint estimation of hand and object poses. The physical constraints mainly focus on enhancing the stability of grasping, which is the most common interaction manner between hands and objects. Together with the physical affinity loss, a context-aware graph network is also proposed to jointly learn independent geometry prior and interaction messages. The whole pipeline consists of two components. First an image encoder is used to predict 2D keypoints from RGB image and then a contextual graph module is designed to convert 2D keypoints into 3D estimations. Our graph module treats the keypoints of hands and objects as two sub-graphs and estimates initial 3D coordinates according to their topology structure separately. Then the two sub-graphs are merged into a whole graph to capture the interaction information and further refine the 3D estimation results. Experimental results show that both our physical affinity loss and our context-aware graph network can effectively capture the relationship and improve the accuracy of 3D pose estimation.

CCS CONCEPTS

• **Computing methodologies** → *Interest point and salient region detections*; *Reconstruction*; Activity recognition and understanding.

KEYWORDS

hand-object pose estimation; grasp stability; contextual graph network

ACM Reference Format:
Nan Zhuang, Yadong Mu. 2021. Joint Hand-Object Pose Estimation with Differentiably-Learned Physical Contact Point Analysis. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463648

*corresponding author.

ICMR '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463648

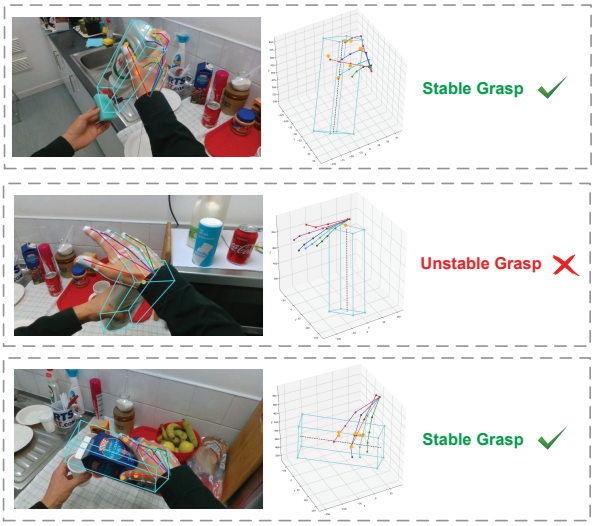

Figure 1: Illustration of stable and unstable grasping. During the interaction, stability is the crucial metric for the hand-object system. This work aims to develop learnable explicit physical constraints which encourage the stability.

1 INTRODUCTION

Human primarily uses hands to interact with the world. Accurate recognition of hand pose [7, 28, 36, 43–45] and the interaction between hands and objects [1, 35, 38] are critical for understanding human activities. It has huge potentials for a wide applications in computer vision, such as augmented reality, human-computer interaction, robotics, which is beneficial to blur the boundaries between the real and virtual worlds. However, accurate detecting 3D poses of hands and the objects that are being handled is a quite challenging task. First, hands move quickly when they are interacting with the world and handling objects, which results in self-occlusions of hands and mutual occlusions between hands / objects from nearly any given point of view. Second, the estimation of 3D hand pose itself is not easy, as hand poses have a high degree of freedom compared to other 3D tasks, such as human pose. Third, hand-object interaction video is usually collected from moving, egocentric cameras (*e.g.*, for Augmented Reality applications), generating a large degree of unpredictable camera motion.

To tackle the hand-object pose estimation task, [38] estimates the pose of hands and objects separately. However, such approach

ignores the correlation between the poses of hand and handled objects, which can be of great significance, as the poses of the hand and the object influence each other. On one hand, estimating the pose of the hand can provide clues for the category and pose of the object. On the other hand, the shape of the object limits the pose of the hand holding it. Recently, a few works [8, 37, 38] directly learn to jointly estimate hand and object poses from RGB images and achieve impressive results. Tekin *et al.* [37] propose a single 3D YOLO model to jointly predict the 3D hand pose and object pose. Similar to 3D human pose estimation [25], Doosti *et al.* propose HOPE-Net [8], which first estimates 2D coordinates of the hand joints and object corners and converts 2D coordinates to 3D space with an Adaptive Graph U-Net.

However, most of these methods are limited by the following factors: Firstly, they do not consider any explicit physical constraints between hand and object poses. Interactions impose constraints on relative configurations of hands and objects naturally. For instance, a stable grasp usually requires force-closure constraint [10, 23, 29], which means that any motion of the object is resisted by a contact force and the object can not break contact with the finger tips without some non-zero external work. Secondly, they usually treat hand keypoints and object keypoints equally, without distinguishing their difference, which can be critical as the position and motion of keypoints in hands and objects can vary a lot.

Our method aims at tackling all these issues. To this end, we propose a learnable physical loss as a regularization and constraints of hand-object poses to enhance their physical stability. Our key insight is, in a stable grasp, the contact points between hands and objects should distribute diversely around the object surface and the center of the hand and the manipulated object should be close possibly. In such a case the hand-object system can be robust against some external forces, such as the object's gravity itself. Contact points will provide enough force to resist object's motion and disturbance. Furthermore, a context-aware graph network for simultaneously predicting 3D hand and object poses is also proposed. Following [8], the whole pipeline is broken into two steps. First an image encoder is utilized to extract image features and predict 2D keypoints for both hands and objects. Second a context-aware graph module is exploited to convert 2D coordinates into 3D coordinates. During the 2D-to-3D conversion, the hand keypoints and object keypoints are first treated as two sub-graph to extract their own structure information and then the two sub-graphs are merged into a whole-graph to pass their interaction messages.

In brief, the main contributions of this paper are summarized as the following.

(1) We first propose a novel learnable physical affinity loss to regularize hand-object poses, which works as an approximate physical constraint to enhance the grasping stability in hand-object pose estimation task.

(2) We develop a context-aware network structure to explore both independent topology structure and mutual interaction messages of hands and objects from the input RGB image, which helps improve accuracy a lot.

(3) Through extensive experiments, we show that our approach can outperform the state-of-the-art methods in a certain margin for the joint hand-object pose estimation task on realistic benchmarks.

2 RELATED WORKS

2.1 Hand Pose Estimation

3D hand pose estimation is a long-standing problem in computer vision domain, and various methods have been proposed. We mainly focus on the more recent deep learning based approaches.

A large body of the works on hand pose estimation operate on depth images as input, which greatly reduces the depth ambiguity of the task. These methods can be further classified into regression-based methods, detection-based methods and hierarchical and structured methods. Regression-based methods [6, 13, 15, 20, 30, 31, 41] aim at directly regressing 3D hand pose parameters such as 3D coordinates or joint angles from the input. Ge *et al.* apply 3D CNNs [15] and PointNet [13] for estimating 3D hand poses directly. However, regressing coordinates from images or point clouds is a highly non-linear problem, which can be hard to learn. Thus detection-based [11, 14, 17, 26, 40] methods work in a dense local prediction manner via setting a heat-map for each keypoint. Moon *et al.* [26] propose a Voxel-to-Voxel prediction network (V2V) for both 3D hand and human pose estimation. Wan *et al.* [40] and Ge *et al.* [17] formulate 3D hand pose as 3D heat-maps and unit vector fields, and estimate these parameters by dense pixel-wise or point-wise regression respectively. Hierarchical and structured methods [5, 9, 24, 30, 31, 47] aim at incorporating hand part correlations or pose constraints into the model.

However, it is relatively difficult to require depth-sensors in daily life. Thus in recent years, there is an obvious trend shifting towards RGB-based solutions [2, 3, 16, 27, 42, 46, 48], which are often less restricted in real-world applications. But the ambiguities in single RGB camera and the lack of texture features make current techniques still far more ubiquitous than depth-based methods. Besides, due to the huge difficulty in accurate 3D annotations for RGB images, most of these works rely on synthetic data as well.

2.2 Hand-Object Pose Estimation

Different from estimating 3D hand pose only, hand-object pose estimation jointly detects the poses of hands and the manipulated objects. Oikonomidis*et al.* [33] treat hand-object interaction as context to better estimate the 2D hand pose from multi-view images. Choi *et al.* [7] train two networks, one object-centered and one hand-centered, to capture information from both the object and hand perspectives, and share information between the two networks to learn a better representation for predicting 3D pose. Panteleris *et al.* [34] generate 3D hand pose and 3D models of unknown objects based on hand-object interactions and depth information. Oberweger *et al.* [32] propose an iterative approach by using Spatial Transformer Networks (STNs) to separately focus on the manipulated object and the hand to predict their corresponding poses. Later they estimate the hand and object depth images and fuse them using an inverse STN. The synthesized depth images are further used to refine the hand and object pose estimations.

Recently, Tekin *et al.* [37] and Doosti *et al.* [8] use deep neural network to estimate hand and object poses from a single RGB image in real-data. Tekin *et al.* [37] propose a single 3D YOLO model to jointly predict the 3D hand pose and object pose. Doosti *et al.* [8] propose HOPE-Net, which first estimates 2D coordinates of the hand joints and object corners and convert 2D coordinates to 3D

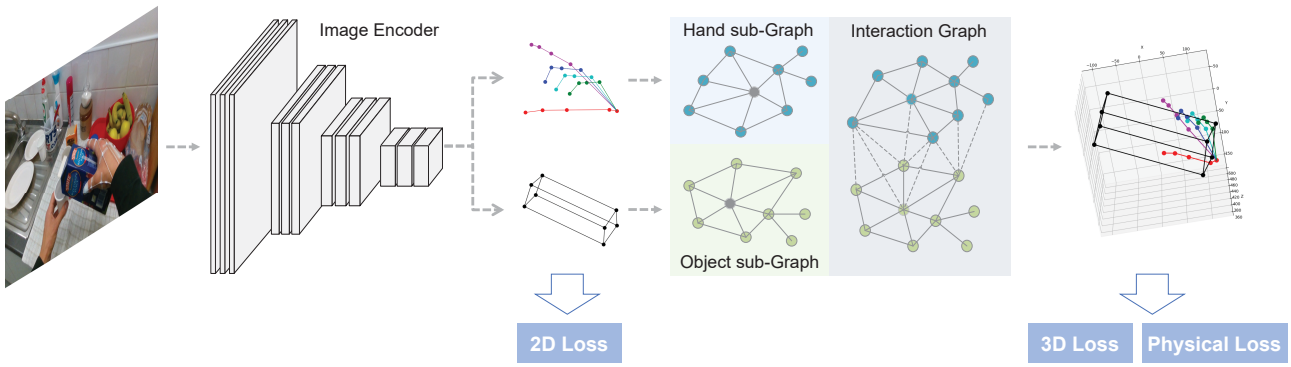

Figure 2: The whole pipeline of our model. First a backbone is used to estimate 2D coordinates of both hand and object keypoints from RGB images. Then context-aware graph module converts 2D coordinates into 3D coordinates for hand and object separately to learn their shape prior and then refine 3D poses jointly according to their interaction messages.

with a Graph U-Net. However, they ignore the physical constraints and affinity between hands and objects during their interaction, which is of great significance.

Hasson *et al.* [18] shows that by incorporating physical constraints, two separate networks responsible for learning object and hand representations can be combined to generate better 3D hand and object shapes, which is similar to our work. However, ours differs from Hasson *et al.* [18] in the following aspects: First they work on a synthetic dataset which is less complex than realistic data. Second they focus on the reconstruction of 3D mesh, which needs more accurate and complex labels. Third, their physical constraints are, when grasping objects, the contact points in hands and objects should be as close as possible but they should not be inside each other at the same time. However, our physical affinity loss considers the stability of grasping from the perspective of forces and mechanics.

3 METHODOLOGY

Given a single image I, we aim at estimating accurate 3D poses of both hands and manipulated objects. In this section, we first give a brief introduction to Graph Convolution Network (GCN) [22], which works as the core component of our model. Then the whole pipeline and network architecture are introduced. In the last, we will present our physical affinity loss in detail.

3.1 Revisiting Graph Convolution

Graph convolution network (GCN) is adopted as the core of our model for its ability to learn high-level representations of relationships between the nodes of graph-based data. Compared with traditional CNN, GCN has its unique convolutional operators for irregular data structures.

Given an input graph with N nodes, k input features and l output features for each node, a graph convolution layer can be defined as,

$$Y = \sigma(\tilde{A}XW), \tag{1}$$

where σ is the activation function, $W \in \mathbb{R}^{k\times l}$ is the trainable weights matrix, $X \in \mathbb{R}^{N\times k}$ is the matrix of input features, $\tilde{A} \in \mathbb{R}^{N\times N}$ is the re-normalized adjacency matrix of the graph as mentioned in [22],

$$\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}, \tag{2}$$

where $\hat{A} = A+I$ and $\hat{D}$ is the diagonal node degree matrix. $\tilde{A}$ simply defines the extent to which each node uses other nodes' features and $\tilde{A}X$ is a new feature matrix in which each node's feature are the average of the node itself and its adjacent nodes. A is the adjacency matrix of graph, which can both be defined by users or learned from training.

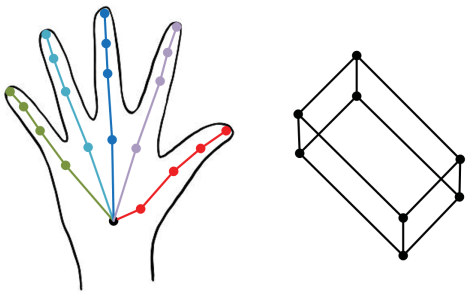

Figure 3: Kinematic structure of hands and objects.

3.2 Network Architecture

Motivated by recent advances in 3D human pose estimation [25] and hand-object pose estimation [8], our model consists of two modules, which first estimates 2D poses of both hands and objects from a single RGB image, and then lift 2D locations to 3D space. The whole pipeline can be seen in Figure 2.

In the first module, an image encoder based on Resnet50 backbone [19] is utilized to extract visual features and predicts initial 2D keypoint locations for both hands and objects. For each hand or object keypoint, its initial 2D coordinate is concatenated with the global image feature and obtain a 2050D feature as its node representation (initial x-coordinate and y-coordinate plus a 2048D

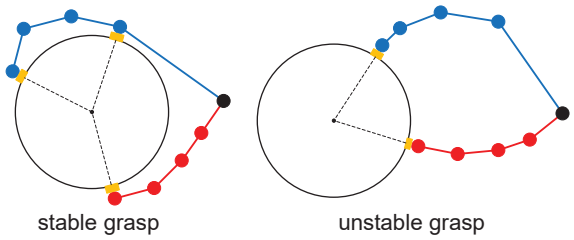

Figure 4: Stable and unstable grasp. Yellow areas are contact points between hands and objects. When the contact points distribute diversely around the object, the grasp can be more stable.

image feature). Then a lightweight 3-layer GCN is applied to refine the initial 2D points, where the adjacency matrix is learned from data.

In the second module, given the estimated 2D keypoints as input, a graph module further learns to convert 2D coordinates into 3D space. Different from Hope-Net [8] which utilizes a Graph U-Net and treats all keypoints in the same manner, we design a context-aware graph module. Our key insight is that hands and objects have their own topology structure. At the same time, there also exists mutual constraints when hands are interacting with objects. Thus we first use two sub-graph networks to capture their own structure for hands and objects separately. And then the two sub-graphs are merged to a whole graph to mine the interaction messages. In this manner, our graph model can explicitly learn separate topology structures and pass interaction messages. The connections of hand and object sub-graph network are defined as their kinematic structure in Figure 3 and their interaction connections are learned from data. The sub-graph network is made up of 3 GCN layers while the merged graph consists of 6 GCN layers.

3.3 Physical Affinity Loss

In this section we will present our physical affinity loss in detail. So far, the prediction of hands and objects does not leverage explicit constraints that guide the hand-object interaction in the physical world. As we know, contacts occur at the surface between the object and the hand, especially for grasping. Hasson *et al.* [18] leverage the fact that the surface of the object and the hand should be as close as possible but they can not interpenetrate each other. However, that is not enough for a stable grasp. In a stable grasp, any motion of the object is resisted by a contact force and the object can not break contact with the finger tips without some non-zero external work, which is called a force-closure grasp. But the judging of a force-closure grasp is a bit difficult. It needs accurate modeling of the location of contact points and the coefficient of friction, which can be hard to obtain in hand-object estimation task. In this paper, motivated by force-closure judgement and zero moment point [39] in robotics, we simplify the physical constraints and propose a differential loss function.

Our motivation is to enhance the stability of grasping from the perspective of forces. Similar to zero moment point [39] in robotics, for a stable grasp, the contact points should distribute

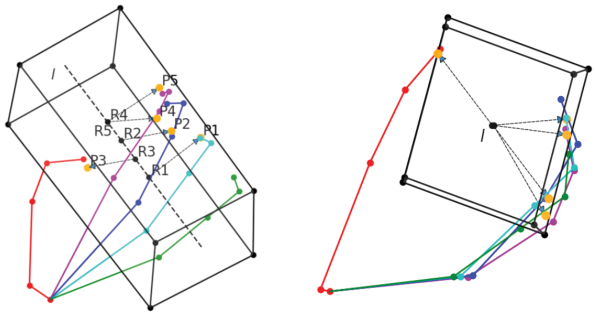

Figure 5: Physical Affinity loss illustration. Yellow points (P1-P5) are contact points between the hand and object. Black points (R1-R5) are the projection of contact points in object's central axis.

around the central axis of the object diversely so that the center of these contact points can locate closer to the center of objects. In such a case contact points can provide enough support forces against any external disturbance, as shown in Figure 4.

Thus we design our physical affinity loss in the following fashion. In brief we first sample the approximate contact points between the hand and object, then obtain direction vectors pointed from the object central axis to these contact points. Our physical affinity loss is defined as a diversity loss based on the cosine similarities among these direction vectors.

Given hand keypoints $V_{Hand} \in \mathbb{R}^{N_h \times 3}$ and object keypoints $V_{Obj} \in \mathbb{R}^{N_o \times 3}$, where $N_h = 21$ and $N_o = 8$ represent the number of keypoints in hand and object separately. We denote the i-th hand keypoint as H_i. As the bounding box of object is approximately a cuboid, it has 8 keypoints (vertexes) and 6 faces. We denote the j-th object face as Π_j. For each face Π_j, we uniformally sample N points along the direction of length and width separately through bilinear interpolation according to the four vertexes and get totally $N \times N$ face points as a face point set $F_j \in \mathbb{R}^{N^2 \times 3}$, $j = 1, 2, 3, \cdots, 6$.

Then for each hand keypoint H_i and each object face Π_j, we can calculate their distance $d(H_i, \Pi_j)$ and support point $supp(H_i, \Pi_j)$ in the following manner:

$$\begin{aligned} d(H_i, \Pi_j) &= \min_{P \in F_j} \|H_i - P\|_2 , \\ supp(H_i, \Pi_j) &= \arg\min_{P \in F_j} \|H_i - P\|_2 , \end{aligned} \tag{3}$$

where the support point between hand keypoint H_i and object face Π_j is defined as the point in face Π_j with the minimal distance to hand keypoint H_i.

From all these support points $\{supp(H_i, \Pi_j)\}$, we select top N_c points with the minimal distances to their corresponding hand keypoints as candidate contact point set C:

$$C = \{supp(H_i, \Pi_j) \mid d(H_i, \Pi_j) \in TopK\{d(H_i, \Pi_j)\}\}. \tag{4}$$

However, note that contacts may not occur between hands and objects, as they are not in touching with each other all the time. For such a case, we use the the radius of objects as a condition to filter the candidate contact points C and get valid contact points C'.

Algorithm 1 The computational pipeline of our method.

Input:
Hand keypoints, V_{Hand};
Object keypoints, V_{Obj};
Output:
Physical Affinity Loss, $L_{physical}$;
for each object face Π_j **do**
  Sample face point set F_j;
  for each hand keypoint H_i **do**
    Calculate $d(H_i, \Pi_j)$ and $supp(H_i, \Pi_j)$;
  end for
end for
Select candidate contact point set C according to Equation 4;
Filter valid contact point set C'' according to Equation 5;
Obtain direction vectors $\{v_i\}$ for points in set C';
Calculate $L_{physical}(V_{Hand}, V_{Obj})$ according to Equation 6;
return $L_{physical}(V_{Hand}, V_{Obj})$;

The radius of an object L is defined as the mean of side lengths of object bounding box's upper surface and lower surface. If the radio of the corresponding distance of a candidate contact point and the object radius is larger than a threshold η, we argue that no contacts occur at this candidate contact point and remove it from candidate point set:

$$C' = \{supp(H_i, \Pi_j) \mid supp(H_i, \Pi_j) \in C \quad and\ d(H_i, \Pi_j) \leq \eta L\}, \tag{5}$$

where we set $N_c = 10$ and $\eta = 0.2$ in our experiments.

After selecting all valid contact points, we get contact point set C' with N_c' contact points. To make a grasp stable, we hope these contact points can distribute diversely around the object central axis l, which is defined as the line between the centers of object bounding box's upper surface and lower surface. Thus for each contact point $P_i \in C'$, we denote its projection on object central axis l as R_i, and define its direction vector as v_i, which is the vector pointed from R_i to P_i, as can be seen in Figure 5. Our physical affinity loss is defined as a diversity loss based on cosine similarity of these direction vectors:

$$L_{physical} = \frac{1}{Z} \sum_{i=1}^{N_c'-1} \sum_{j=i+1}^{N_c'} \frac{v_i \cdot v_j}{\|v_i\|_2 \left\|v_j\right\|_2}, \tag{6}$$

where $Z = \frac{1}{2} N_c'(N_c' - 1)$ is the normalization factor. The calculation of our physical affinity loss can be summarized in Algorithm 1.

3.4 Full Objective

The loss function to train our network is composed of two parts, fully supervised loss and our physical affinity loss. The supervised loss is defined as the Mean Squared Error for both 2D coordinates and 3D coordinates between predictions and ground-truths, and physical affinity loss is defined in Section 3.3,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{2D} + \lambda_2 \mathcal{L}_{3D} + \lambda_3 \mathcal{L}_{physical}, \tag{7}$$

where $\lambda_1 = 0.01$, $\lambda_2 = 1$ and $\lambda_3 = 10$ for training.

4 EXPERIMENTS

4.1 Datasets

To evaluate effectiveness of our network and our physical affinity loss, we perform experiments on a realistic dataset: First-Person Hand Action (FPHA) [12]. Besides, following [8], we also pre-train our network on a synthetic dataset Obman [18] for better performance. All of these datasets use 21 joints model for hands and 8 joints for object bounding boxes.

First-Person Hand Action [12] contains first- person videos of hand actions performed on a variety of objects. The dataset consists of 1,175 gesture videos with 45 gesture classes. The videos are performed by 6 actors under 3 different scenarios. A total of 105,459 video frames are annotated with accurate hand pose and action classes. Both 2D and 3D annotations of the total 21 hand keypoints are provided for each frame. However, only a small subset with 21,501 frames include 6D object pose annotations, which we denote as FPHA-HO in the following paragraphs. The objects in FPHA-HO are *milk, juice bottle, liquid soap, and salt*, and actions include *open, close, pour, and put*. We conduct our experiments on FPHA-HO, with 11,019 frames for training, 5,442 frames for validation and 5,040 frames for testing.

Obman [18] is a large dataset of synthetically-generated images of hand-object interactions. Images in this dataset are produced by rendering meshes of hands with selected objects from ShapeNet [4]. The dataset consists of 141,550 frames for training, 6,463 for validation and 6,285 for testing. Despite the large-scale of the annotated data, the models trained with these synthetic images do not generalize well to real images [8]. Nevertheless, it is still helpful to pre-train our model on the large-scale data of ObMan, and then fine-tune using real images of FHAD-HO.

4.2 Evaluation Metrics

Following [8, 37], we use the percentage of correct keypoint estimates (3D PCK) and percentage of correct poses for 3D hand pose estimation and 6D object pose estimation respectively. We consider a hand pose estimate to be correct when the mean distance between the predicted and ground-truth joint positions is below a certain threshold (in mm). When using the percentage of correct poses to evaluate 6D object pose estimation, we take a pose estimate to be correct if the 2D projection error is less than a certain threshold (in pixels).

4.3 Implementation Details

We implement our method with the PyTorch framework, and optimize the objective function with the Adam optimizer [21] with mini-batches of size 256. All experiments are conducted on a single server with four NVIDIA TITAN GPUs. Following [8], we pre-train the whole network on Obman [18] before training on FHAD-HO [12].

Specifically, on each dataset, we first train the graph module which converts 2D coordinates to 3D space and the image encoder that predict 2D keypoints separately and then fine-tune the whole pipeline in an end-to-end manner. For training graph module, we use the ground-truth 2D coordinates as the input and train this module for 10,000 epochs with an initial learning rate of 0.001 and multiply it by 0.1 every 4000 epochs. Similarly, the Image Encoder

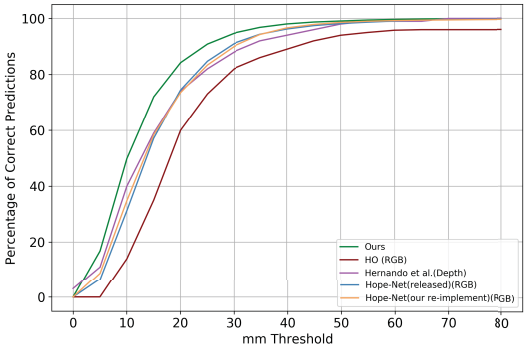

Figure 6: The percentage of correct 3D hand pose of our model on FPHA-HO using different thresholds (in mm).

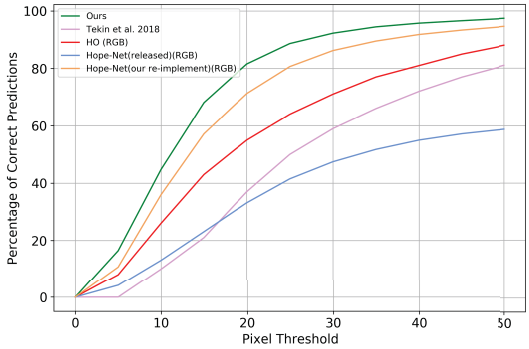

Figure 7: The percentage of correct 2D object pose of our model on FPHA-HO using different thresholds (in pixel).

is trained for 1,000 epochs with an initial learning rate of 0.001 and multiplied by 0.9 every 100 epochs to predict 2D keypoints. Finally, the two modules are fine-tuned in an end-to-end manner with a learning rate of 0.0001 for 100 epochs. During training, all images are resized to 224×224 pixels.

4.4 Experimental Results

We now report the performance of our model on FPHA-HO and compare our results to the state-of-the-art results of [8] [37].

In Table 1 we present the mean 3D distance of hand keypoints and object keypoints in mm. As we can see, our results obviously outperform the other two methods in a certain margin. Please note that the results of HO [37] are reported from their original paper, while the results of Hope-Net [8] are our re-implementation results according to publicly available codes, which is slightly better than the authors' released model (in which the mean error of hand and object are 16.14 mm and 72.89 mm separately).

Network	HO[37]	Hope-Net[8]	Ours
Hand Pose error (mm)	15.81	15.04	**12.53**
Object Pose error (mm)	24.89	21.33	**19.09**

Table 1: Quantitative 3D errors(mm) of predicted hand and object poses.

Figure 6 presents the percentage of correct 3D hand poses for various thresholds (measured in millimeters) in detail. The results show that our method not only outperforms RGB-based models like Hope-Net [8] and HO [37], but also beats some depth-based model [12], without any depth information or temporal information. Our method shows good superiority for smaller thresholds (from 0 to 40 mm). Figure 7 presents the percentage of correct object pose for various pixel thresholds for the 2D projection. Our method is slightly better than our re-implemented Hope-Net [8], and outperforms HO [37] and Tekin *et al.* [38] in a certain margin.

In Figure 8 we show some visual examples of both our method and Hope-Net [8]. We see that our method can better capture the geometry shape, especially for object shapes. Even in some extreme cases like the second column in Figure 8, the hands and objects are beyond the image boundary, our estimation of object pose is more accurate and more like a cuboid. We believe this is due to our specific design which distinguishes hands and objects first before capturing their interaction and learns better prior knowledge of hands and objects separately. Besides, our results also show better affinity between hands and objects as our physical loss encourages more stable situations (like the first and the third case in Figure 8).

More examples are presented in Figure 10 for some other objects and actions. We can see that for some scenes where several keypoints are beyond the image boundary slightly, our method can still make accurate predictions according to the learned prior knowledge of hand and object geometry information.

4.5 Ablation Study

In this section we also conduct ablation studies to identify the importance of each components for achieving our results.

Network	Hand Pose error	Object Pose error
Adaptive Graph U-Net	14.36	21.65
Hand Sub-graph Only	12.22	-
Object Sub-graph Only	-	14.41
Full Graph	**10.38**	**13.85**

Table 2: Mean error (measured in millimeter) on 3D hand and object pose estimation with 2D ground-truth as inputs.

Context-aware Graph Module. Our model and Hope-Net [8] share a similar pipeline, which first estimate 2D coordinates of keypoints from RGB images and then convert 2D coordinates to 3D space. The most difference is our 2D-to-3D conversion module utilizes a sub-graph based design, while Hope-Net uses Adaptive Graph U-Net. To show the effectiveness of our sub-graph structure, we directly compare it with Adaptive Graph U-Net. All these modules are trained to lift 2D coordinates of hand and object keypoints to 3D space. We directly use the 2D ground-truth coordinates as input and Mean Square Error as loss function to show their ability. And all the results follow the same training procedure which first

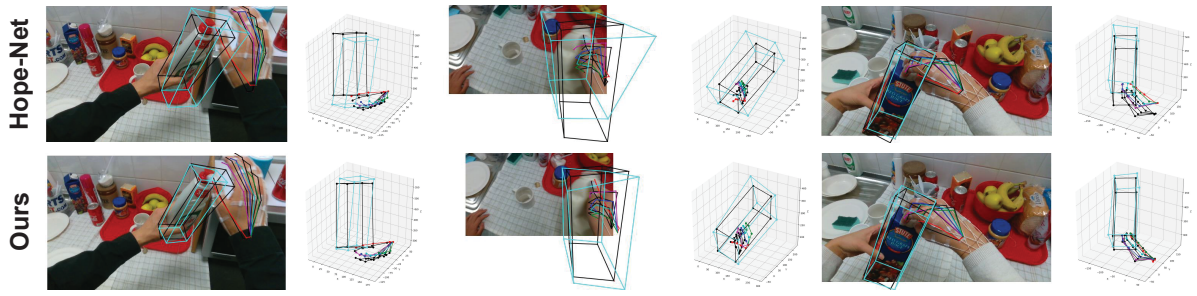

Figure 8: Comparison with previous state-of-the-art method Hope-Net [8]. Our method can better capture the geometry shape of hands and objects, even in some extreme scenes, such as hands and objects are partly out of the image boundary.

pre-train on synthetic data and then fine-tune on realistic data. The quantitative results can be seen in Table 2. Note that the results of Graph U-Net are our re-implementation according to publicly available codes, which are also better than the released model (where hand error and object error are 22.07 mm and 43.76 mm separately).

As we can see, our full graph model is significantly better than other methods on both hand pose and object pose estimation. Meanwhile, hand sub-graph only and object sub-graph only also outperform Adaptive Graph U-Net. Especially for object pose estimation, a large margin can be observed between Adaptive Graph U-Net and Object Sub-graph Network. We believe it is due to the following reasons: Adaptive Graph U-Net treats each keypoint equally, no matter from hands or objects. Such a treatment ignores the factor that hands and objects have prior geometry knowledge and the position and motion of keypoints in hands and objects can vary a lot during their interaction. However, our sub-graph based modules fully consider the topology structure of hands and objects themselves as well as their interactions. Thus our context-aware graph module can successfully combine the local structure knowledge and mutual interaction information.

Loss Function	Hand Pose error	Object Pose error
MSE	12.83	20.83
MSE + Physical loss	**12.53**	**19.09**

Table 3: Mean error (measured in millimeter) on 3D hand and object pose estimation with different loss functions.

Physical Affinity Loss. We further study the effect of our physical affinity loss as a fine-tune step. Quantitative results can be seen in Table 3. By introducing our physical affinity loss and further fine-tune the whole model, we can see that the mean 3D error of object pose achieves a certain improvement. Besides, the results of hand pose also have a slightly improvement as well. This is mainly because that our physical affinity loss explicitly restricts and limits the relationship between the hand and object poses, encouraging their positions and shapes meet the constraints for grasp stability and be in affinity with each other.

In Figure 11 we further visualize some examples of contact points in FPHA-HO dataset. As we can see, our algorithm effectively detects these contact locations and make these contacts distribute diversely around the object central axis if contact exists.

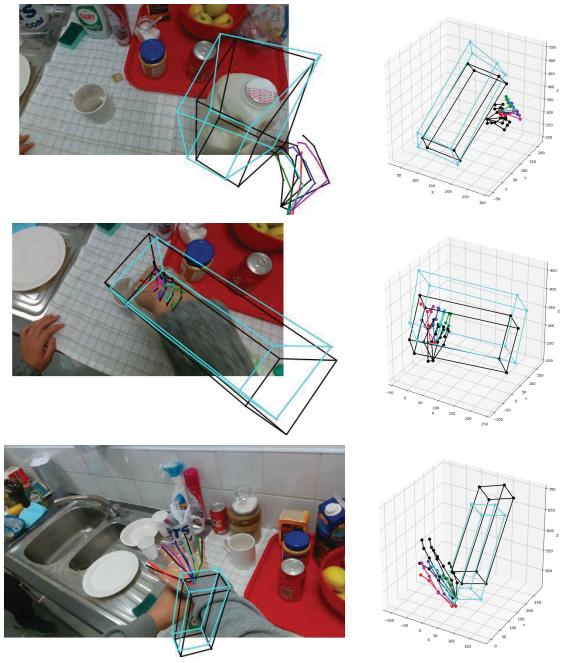

Figure 9: Some failure cases.

5 FAILURE CASES

We show some difficult cases in Figure 9, the estimations of which are clearly unsatisfactory.

For example, in the first and second row of Figure 9, we can see significant errors primarily caused by out-of-sight hand or object parts. In the third row, the object is totally occluded by the hand. Although the object and hand shapes are approximately correct compared to the ground-truths, there still exist obvious offsets in 3D space for these hard examples.

6 CONCLUSION

In this paper, we propose a learnable physical loss together with a context-aware graph model for jointly hand-object pose estimation from a single RGB image. Our physical affinity loss encourages

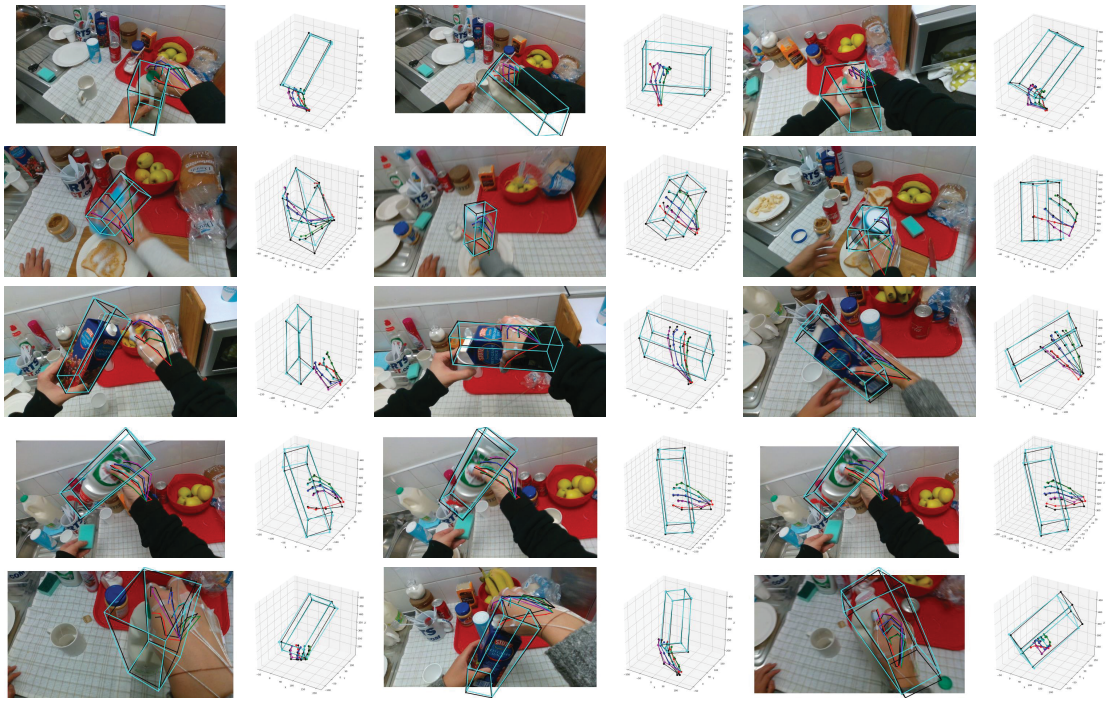

Figure 10: Some visual examples. Colored skeletons are our predictions and black ones are ground-truth. The proposed method can handle various objects and actions. Even when some keypoints are out of the image boundary, our method can still make predictions according to learned prior knowledge of hand and object geometry information.

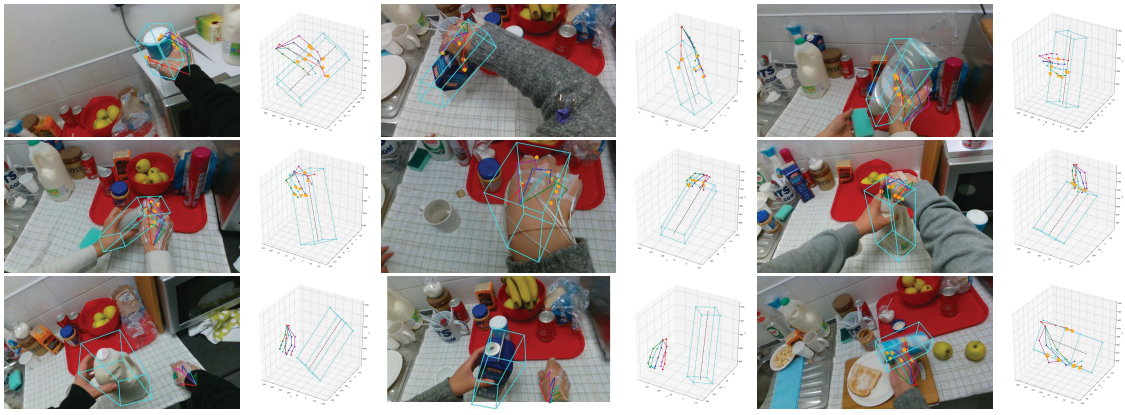

Figure 11: Visual examples of contact points between hands and objects. Yellow areas and points are our detected contact areas.

stable grasping and works as a effective regularization. Our context-aware model works as a "local-to-global" architecture and explores both the independent structure and mutual interaction of hands and objects. Future work could further take temporal information into consideration to both improve pose estimation results and understand human actions.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China (2020AAA0104400), National Natural Science Foundation of China (61772037) and Beijing Natural Science Foundation (Z190001).

REFERENCES

[1] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. 2016. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image. In *CVPR*.

[2] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. 2018. Weakly-Supervised 3D Hand Pose Estimation from Monocular RGB Images. In *ECCV*.

[3] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat-Thalmann. 2019. Exploiting Spatial-Temporal Relationships for 3D Pose Estimation via Graph Convolutional Networks. In *ICCV*.

[4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. *CoRR* abs/1512.03012 (2015).

[5] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. 2020. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing* 395 (2020), 138–149.

[6] Xinghao Chen, Guijin Wang, Cairong Zhang, Tae-Kyun Kim, and Xiangyang Ji. 2018. SHPR-Net: Deep Semantic Hand Pose Regression From Point Clouds. *IEEE Access* 6 (2018), 43425–43439.

[7] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. 2017. Robust Hand Pose Estimation during the Interaction with an Unknown Object. In *ICCV*.

[8] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. 2020. HOPE-Net: A Graph-Based Model for Hand-Object Pose Estimation. In *CVPR*.

[9] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. 2019. CrossInfoNet: Multi-Task Information Sharing Based Hand Pose Estimation. In *CVPR*.

[10] Haoshu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. 2020. GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping. In *CVPR*.

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

[12] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. 2018. First-Person Hand Action Benchmark With RGB-D Videos and 3D Hand Pose Annotations. In *CVPR*.

[13] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. 2018. Hand PointNet: 3D Hand Pose Estimation Using Point Sets. In *CVPR*.

[14] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2016. Robust 3D Hand Pose Estimation in Single Depth Images: From Single-View CNN to Multi-View CNNs. In *CVPR*.

[15] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2017. 3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation from Single Depth Images. In *CVPR*.

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

[17] Liuhao Ge, Zhou Ren, and Junsong Yuan. 2018. Point-to-Point Regression PointNet for 3D Hand Pose Estimation. In *ECCV*.

[18] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. 2019. Learning Joint Reconstruction of Hands and Manipulated Objects. In *CVPR*.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[20] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. 2020. Hand-Transformer: Non-Autoregressive Structured Modeling for 3D Hand Pose Estimation. In *ECCV*.

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

[22] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[23] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. 2019. PointNetGPD: Detecting Grasp Configurations from Point Sets. In *ICRA*.

[24] Meysam Madadi, Sergio Escalera, Xavier Baró, and Jordi Gonzàlez. 2017. End-to-end Global to Local CNN Learning for Hand Pose Recovery in Depth data. *CoRR* abs/1705.09606 (2017).

[25] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In *ICCV*.

[26] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. 2018. V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation From a Single Depth Map. In *CVPR*.

[27] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated Hands for Real-Time 3D Hand Tracking From Monocular RGB. In *CVPR*.

[28] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-Time Hand Tracking under Occlusion from an Egocentric RGB-D Sensor. In *ICCV*.

[29] Van-Duc Nguyen. 1988. Constructing Force-Closure Grasps. *Int. J. Robotics Res.* 7, 3 (1988), 3–16.

[30] Markus Oberweger and Vincent Lepetit. 2017. DeepPrior++: Improving Fast and Accurate 3D Hand Pose Estimation. In *ICCV*.

[31] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. 2015. Hands Deep in Deep Learning for Hand Pose Estimation. *CoRR* abs/1502.06807 (2015).

[32] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. 2020. Generalized Feedback Loop for Joint Hand-Object Pose Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 8 (2020), 1898–1912.

[33] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. 2011. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*.

[34] Paschalis Panteleris, Nikolaos Kyriazis, and Antonis A. Argyros. 2015. 3D Tracking of Human Hands in Interaction with Unknown Objects. In *BMVC*.

[35] Mahdi Rad and Vincent Lepetit. 2017. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth. In *ICCV*.

[36] Grégory Rogez, James Steven Supancic III, and Deva Ramanan. 2015. First-person pose recognition using egocentric workspaces. In *CVPR*.

[37] Bugra Tekin, Federica Bogo, and Marc Pollefeys. 2019. H+O: Unified Egocentric Recognition of 3D Hand-Object Poses and Interactions. In *CVPR*.

[38] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. 2018. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*.

[39] Miomir Vukobratovic and Branislav Borovac. 2004. Zero-Moment Point - Thirty Five Years of its Life. *Int. J. Humanoid Robotics* 1, 1 (2004), 157–173.

[40] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2018. Dense 3D Regression for Hand Pose Estimation. In *CVPR*.

[41] Guijin Wang, Xinghao Chen, Hengkai Guo, and Cairong Zhang. 2018. Region ensemble network: Towards good practices for deep 3D hand pose estimation. *J. Vis. Commun. Image Represent.* 55 (2018), 404–414.

[42] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. 2019. Aligning Latent Spaces for 3D Hand Pose Estimation. In *ICCV*.

[43] Qi Ye and Tae-Kyun Kim. 2018. Occlusion-Aware Hand Pose Estimation Using Hierarchical Mixture Density Network. In *ECCV*.

[44] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis A. Argyros, and Tae-Kyun Kim. 2018. Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals. In *CVPR*.

[45] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. 2017. BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis. In *CVPR*.

[46] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. 2019. End-to-End Hand Mesh Recovery From a Monocular RGB Image. In *ICCV*.

[47] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. 2016. Model-Based Deep Hand Pose Estimation. In *IJCAI*.

[48] Christian Zimmermann and Thomas Brox. 2017. Learning to Estimate 3D Hand Pose from Single RGB Images. In *ICCV*.