Parallel mesh reconstruction streams for pose estimation of interacting hands

Uri Wollner
Artificial Intelligence, GE Research
uri.wollner@ge.com

Guy Ben-Yosef
Artificial Intelligence, GE Research
guy.ben-yosef@ge.com

Abstract

We present a new multi-stream 3D mesh reconstruction network (MSMR-Net) for hand pose estimation from a single RGB image. Our model consists of an image encoder followed by a mesh-convolution decoder composed of connected graph convolution layers. In contrast to previous models that form a single mesh decoding path, our decoder network incorporates multiple cross-resolution trajectories that are executed in parallel. Thus, global and local information are shared to form rich decoding representations at minor additional parameter cost compared to the single trajectory network. We demonstrate the effectiveness of our method in hand-hand and hand-object interaction scenarios at various levels of interaction. To evaluate the former scenario, we propose a method to generate RGB images of closely interacting hands. Moreover, we suggest a metric to quantify the degree of interaction and show that close hand interactions are particularly challenging. Experimental results show that the MSMR-Net outperforms existing algorithms on the hand-object FreiHAND dataset as well as on our own hand-hand dataset.

1 Introduction

Human daily behavior include intensive interactions with surrounding objects and with other humans using hands. Capturing such interactions in natural RGB images and analyzing them is therefore important for a wide variety of applications such as monitoring human activity, virtual/augmented reality, and human-machine interactions among others. In particular, analyzing close hand-hand or hand-object interaction scenarios (Fig. 1) in terms of hand and object poses, will eventually allow better visual understanding of the interactions’ type and tone.

Current hand pose estimation models from RGB images predict poses in either 2D, 3D, or both. While traditional approaches were focused on solely estimating coordinates of hand joints and fingertips (e.g., [36, 51, 33, 37]), recent models tend to estimate a 3D dense hand mesh posed at the predicted configuration (e.g., [1, 48, 12, 9, 21, 26, 6, 24]). To develop and test hand pose estimation algorithms, multiple real [36, 21, 8, 52, 26] and synthetic [29, 51, 12, 23] hand datasets were collected and generated in recent years. These datasets encompass various hand-object interactions [41, 21, 11, 12, 39, 2, 8, 52], single-hand gestures [47, 51], as well as hand-hand interactions [41, 26, 23].

Despite the recent improvement in prediction accuracy of hand pose estimation models for hand-object scenarios, their prediction performance for intricate hand-hand and hand-object interactions remain unclear. Furthermore, to the best of our knowledge, a metric to identify these intricate settings does not currently exist. To this end, we develop a novel mesh-based dataset generator for closely-interacting hand-hand and hand-object scenarios. We then investigate the performance of existing models on the generated data. In addition, we propose a visibility-ratio metric to identify and characterize hand interaction scenes.

Based on insights of our analysis, we develop a hand reconstruction model consisting of Graph Convolutional Neural Networks (GCNs). Our GCN decoder architecture is structured with multi-resolution mesh maps that are interconnected to create several mesh decoding trajectories. This structure allows high-resolution mesh layers to ob-
tain global contextual information that exist at the low-resolution mesh layers. Experimental results on external and internal datasets show that our model outperforms recently proposed models on hand mesh reconstruction and pose estimation tasks.

To conclude, our key contributions are as follows:

- We present a hand mesh pose estimation model that utilizes a strategy of multiple mesh decoding paths.
- We introduce a method to generate synthetic examples of close hand-hand and hand-object interactions. It extends previous work (e.g., [12]) by allowing spatial configurations of two or more intertwined articulated objects.
- We introduce a metric to assess the degree of hands’ interactions in an image. We use it to form groups of hand-hand interaction scenarios and test the accuracy of hand pose estimation models on them.

2 Related work

Hand mesh reconstruction. Many traditional 3D hand pose approaches have utilized neural networks to directly regress 3D hand keypoints’ positions [51, 16, 3, 7, 29, 38, 46, 37]. More recent approaches estimate a mesh of the 3D hand surface [21, 50, 9, 48, 26, 6, 24] by either (a) predicting parameters of a deformable 3D hand model, such as the MANO model [35], or (b) estimating 3D coordinates of mesh vertices directly. In both cases, the 3D hand keypoints are obtained by multiplying the estimated mesh by a predefined regression matrix.

Lin et al. [24] presented a multi-layer Transformer [42] encoder with progressive dimensionality reduction to reconstruct a 3D mesh. They show that their model captures short- and long-range interactions among body joints and mesh vertices. This work has motivated us to incorporate a transformer block in our network as well to better capture these long-range relations.

Our work is also inspired by recent papers suggesting the use of Graph Convolution Networks (GCNs) [19, 45] for propagating contextual information along the graph structure. The purpose of GCNs in the context of hand pose estimation algorithms is to contribute to the learning of high-level relationships between hand keypoints or mesh vertices. Typically, the pose model begins with encoding an input image into a latent feature vector. Then, a GCN decodes the latent features into a 3D hand skeleton or mesh.

Ge et al. [9] were the first to include GCNs in a hand pose estimation model. They proposed a model to reconstruct a hand mesh from a latent feature vector that is computed based on 2D hand joints heat-maps and feature-maps obtained from a stacked hourglass network [31]. The reconstruction model uses GCNs that represent locally connected mesh vertices at different refinement levels. The nodes of the GCNs are the mesh vertices and their edges are defined based on the mesh topology. The model starts with a low-resolution spatial map (a coarse mesh) that is upsampled gradually to obtain a high-order one (a finer mesh). Kulon et al. [21] also suggested a GCN-based mesh reconstruction model. Yet, in contrast to [9] that used spectral graph convolution operations in their network, [21] used convolution operations applied to connected mesh nodes according to a local spiral patch operator. Other models include Choi et al. [6] who proposed a GCN-based system to estimate 3D coordinates of hand mesh vertices from 2D hand pose. We note that [21, 9, 6] models are structured with a single, sequential, mesh decoding path. In Sec. 4 we present a GCN-based architecture that uses spiral convolution as well, but incorporates multiple interconnected paths for mesh decoding.

Datasets of synthetic hand images. Datasets and benchmarks with synthesized single hand images have been introduced as an alternative to collecting and annotating real-world data [51, 29]. Images in these datasets were typically generated by a computer graphics software that rendered a digitized hand model. To create a more realistic hand appearance, GANs were further employed on the rendered images [29].

Further work on synthetic hand-object images was done by Hasson et al. [12] who introduced the ObMan (Object Manipulation) dataset. [12] adapted the MANO hand model [35] and combined it with the GraspIt software [25] to generate hand grasp poses interacting with 2.7K everyday object models. The GraspIt algorithm was used to move the hand model towards the object and increase the hand-object contact surface while performing a grasping motion. In its current implementation, the GraspIt algorithm is able to create hand-object interactions such that the object is static and the hand is dynamic,
Figure 1: Closely interacting hands during (a) clasping, (b) stroking, (c) palm reading, and (d) washing. We emulate such scenarios using generated examples that include close hand-hand interactions (e-f), and close hand-object interactions (g-h). In such interactions, a hand’s fingers cross another hand’s fingers or pass through a perforated object. Thus, potentially challenging hand pose estimation algorithms.

i.e. the hand joints can rotate. However, it cannot provide interaction examples where two articulated objects are dynamically moving; a requirement for creating hand-hand interactions. To alleviate this constraint, we present in Sec. 3 a new method to create interactions of two dynamically moving articulated objects. We then use this method to generate synthetic hand-hand dataset for training and evaluating hand pose estimation models.

**Hand-hand interactions.** Very few studies were dedicated for pose estimation of interacting hands, including proposed datasets and models [32, 41, 30, 26, 23]. [26] collected 2.6 million real-captured images of interacting hand poses, and proposed a model for hand pose estimation of a single or two interacting hands. Our current work supplements previously published datasets by creating a large synthetic dataset of closely interacting hands at various configurations. The generated dataset consists of non-intersecting hand meshes that are suitable for development of mesh reconstruction models. Moreover, our dataset extends current mesh-annotated datasets for hand-object interactions. The currently public hand-object interaction datasets contain scenes with objects having simple geometric shapes (e.g., cylindrical or rectangular). Our generator can also produce scenes of hands interacting with articulated objects, in which the hand and object are intertwined (e.g., Fig 1g). In this work we consider them similar to examples of close hand-hand interactions.

**Multiple decoding paths.** Strategies for multiple encoding and decoding paths are widely studied for convolution networks operating on images [40, 31, 5, 44, 43]. These approaches interconnect data maps at different resolution levels that mainly differ in the manner and extent that these connections are integrated. For example, [40] use separated paths that are aggregated at the end of the network process, [31, 5] use connections between encoding and decoding layers at the same resolution level, and [44, 43] use extensive integration between all resolution maps throughout the encoding (or decoding) stage. Similar to the latter approach, we also use multiple connections between differently-sized resolution maps, as shown in Fig. 3. To the best of our knowledge, we present the first multi-resolution interconnections graph convolution network for mesh reconstruction.

### 3 INTER-SH dataset

In this section we describe a novel method for generating INTERacting Synthetic Hand (INTER-SH) examples of close hand-hand and hand-object interactions as captured by an RGB camera. We then use the readily available ground truth annotations in their various forms, such as hand/object masks, to develop and evaluate hand pose predictors.

We simulate hands in our synthetic dataset based on the left and right hands models of MANO [35]. The objects in the hand-object interaction examples are partly taken from the YCB dataset [4] and partly self-generated.

**3.1 Generating Mesh Interactions**

With the mesh models defined in Supplementary Sec. 7.1 and pose generating algorithm in Sec. 7.2, we simulate interactions between two hand meshes, or a hand mesh and an object.

**Textures.** Similar to [1], we assign the RGB values
Figure 2: cylindrical presentation of MANO hand model [35] (a-b) and a chain model (d-e) during data generation routine. The textured mesh scenes of (b) and (e) are shown in (c) and (f), respectively. In (a), the bottom hand is initialized with a random pose. The upper hand is in neutral pose. (b) Final pose of an interacting hands scene upon random motion of finger joints of both hand models. (d) The chain object is static and the hand is its neutral pose. (e) Final pose of hand interacting with chain. Validity of hand poses is based on collision detection of cylindrical rigid bodies.

noted per 3D hand scan collected in [35] to each vertex of the MANO mesh. The RGB assignment is performed based on the closest vertex in the original 3D scan to a vertex in the MANO mesh in neutral pose. The vertices' values are then interpolated along the mesh surface. The textures for the YCB objects [4] are included in the dataset.

**Rendering.** We render the images using Pyrender. For each hand-hand and hand-object configuration, we render object-only, hand-only, and complete scene images, as well as their corresponding segmentation maps.

While our generated hand-object interaction scenes may resemble the output of previously proposed workflows, it is the ability to generate intricate hand-hand and hand-object interactions that contrasts our work from other generated datasets.

4 Multi-path Mesh Decoder

4.1 Architecture

The input to our model is an image of size 224 × 224 centered around a single or interacting hand. The model’s output is a prediction of 3D coordinates of hand mesh vertices \( V \). Our network consists of (i) a Convolutional Neural Network encoder that provides a latent feature vector, (ii) an intermediate module comprising of a fully connected (FC) layer followed by a reshape layer and a Transformer [42] block, and (iii) a multi-scale graph convolution mesh decoder. The purpose of the Transformer block is to promote short- and long-range interactions of the initial coarse mesh representation input to the decoder.

**Multi-path decoder based on graph convolutions.** The implementation of the convolution function in our GCN is done by constructing a fully connected layer such that same linear operation is applied to all features of \( e(v) \), for each \( v \) [22, 21]. More formally, let \( f(e(v)) \) be the sequence of concatenated features for the nodes in \( e(v) \). Let \( g \) be a convolution kernel, and \( f \) the features of all nodes. Then the convolution operation is:

\[
(f * g)_v = \sum_{i=1}^{\vert f(e(v)) \vert} g_i f(e(v))_i.
\]  

where \((f * g)_v\) is the output of the convolution window for \( v \). The convolution is followed by the Exponential Linear Unit activation function that was selected based on empirical evidence. The spatial neighbourhood for the convolution operation were defined using a spiral patch operator similar to [22, 21].

Previously proposed GCN models use a single mesh decoding path. In contrast, we implement a strategy that allow for multiple and parallel mesh decoding trajectories, similar to previous multi-scale fusion works on image encoding (e.g. [44, 43]). The structure of the graph convolution layers and their connections is shown in Fig. 3.

In the proposed model, there are five stages. At each stage, a larger spatial resolution representation is added with respect to the previous stage. Apart from the first stage, all stages comprise of multi-resolution blocks, consisting of a multi-resolution group convolution and a
Figure 3: Architecture of the proposed hand pose estimation model. The input image is passed through a CNN encoder followed by an intermediate module and a GCN-based mesh decoder. Nodes in the GCN are hand vertices. In the proposed model, there are five stages. At each stage, a larger spatial resolution representation is added with respect to the previous stage. The convolutional stages comprise of multi-resolution blocks, consisting of a multi-resolution group convolution and a multi-resolution convolution.

multi-resolution convolution [43]. For the latter convolution operator, a connection between input channels and output channels of differently-sized resolution data is achieved by utilizing precomputed upsampling and downsampling matrices.

To compute the upsampling and downsampling matrices, we construct five coarse mesh representations of the original mesh. At each coarsening stage, the number of vertices are reduced by a factor of 2 [34]. The downsampling matrices of each stage are obtained by iteratively contracting vertex pairs based on quadric error metrics. The vertices of the downsampled mesh are a subset of the original mesh vertices. Vertices discarded during downsampling are projected into the closest triangle of the coarse mesh. Then, the barycentric coordinates of the projected vertex are used to define interpolation weights for the upsampling matrix.

The convolution blocks used within our decoder are structured as a variant of basic residual blocks [14]. Specifically, we replace the 2D convolution layers with graph convolutions defined in Equation 1, and the batch normalization layers with layer normalizations. Data fusion layers of [43] were also incorporated and modified similar to our altered ResNet blocks.

4.2 Training

As loss function we used the L1 norm between the ground truth mesh vertices $V \in \mathbb{R}^{M \times 3}$ to the predicted vertices $\hat{V}$. That is,

$$L = \frac{1}{M} \sum_{j=1}^{M} ||V_j - \hat{V}_j||_1. \quad (2)$$

To train our network we use the Adam solver [18] with learning rate $10^{-4}$ for 200 epochs. Learning rate decay with factor 0.5 occurs after every 50 epochs. The images are normalized with the mean 0.5 and standard deviation of 1.0. We augment the data with random image crops and transformations. We use batch size of 32. Ground truth 3D hand vertices were translated such that the middle metacarpophalangeal joint is at the origin.
Table 1: Performance comparison with published methods, evaluated on FreiHAND online server. Bold faced text imply the best score.

| Backbone | Model                | PA-MPVPE | PA-MPJPE | F@5 mm | F@15 mm |
|----------|----------------------|----------|----------|---------|---------|
| Resnet50 | Hasson et al. [12]   | 13.2     | -        | 0.436   | 0.908   |
| Resnet50 | Kulon et al. [21]    | 8.6      | 8.3      | 0.614   | 0.966   |
| Resnet50 | Doosti et al. [7]    | 9.2      | -        | -       | -       |
|          | Pose2Mesh [6]        | 7.8      | 7.7      | 0.674   | 0.969   |
|          | I2LMeshNet [27]      | 7.6      | 7.4      | 0.681   | 0.973   |
| HRNet64  | METRO [24]           | 6.7      | 6.8      | 0.717   | 0.981   |
| Resnet50 | Ours                 | 7.6      | 7.4      | 0.674   | 0.97    |
| HRNet18  | Ours                 | 7.0      | 7.2      | 0.701   | 0.977   |
| HRNet48  | Ours                 | 6.6      | 6.7      | 0.719   | 0.981   |

5 Experiments

In this section, we present the datasets and corresponding evaluation protocols used to compare our method to the state-of-the-art. In addition, we provide an analysis of our framework and suggest new metrics that help define a set of hard-cases for the hand-pose estimation problem given a RGB image.

5.1 Datasets

Our experiments were limited to datasets that provide ground truth annotations for hand mesh reconstruction. We avoided datasets that only contains joints annotations, since techniques to infer mesh from joint keypoints (such as regressing a MANO model [35]) tend to generate artifacts that harm the training procedure of the model. For hand mesh reconstruction this leaves us with the Friehand datasets [52], which is the main benchmark for hand pose models today, and our own INTER-SH dataset focusing on challenging hand interactions examples.

FreiHAND is a dataset with 130,240 training images [52], including single hands and hand-object interactions, taken with a green screen as a background. The test set, consisting of 3,960 samples, was collected without the green screen in indoor and outdoor environments. The FreiHAND test set annotations is not available. The evaluation of the hand pose predictions is performed through submission of results to an online competition.

5.2 Evaluation Metrics

PA-MPJPE reconstruction error [49] performs a 3D alignment using Procrustes analysis (PA) [10] followed by Mean-Per-Joint-Position-Error (MPJPE) [15] computation. MPJPE measures the average Euclidean distances between the ground truth joints and the predicted joints. In addition, we report the F-score at a given threshold d (F@d) defined as the harmonic mean of precision and recall [20]. Moreover, we compute the percentage of correct points (3D PCK) for different thresholds, and the Area Under Curve (AUC) for PCK.

To further study the effect of the degree of hand-hand interactions on these metrics, we propose to group hand-hand interaction scenarios based on a visibility ratio metric.

Visibility ratio metric. Recall from Sec. 3 that in addition to generating a synthetic image, we also generate separate masks for the hands and objects that appear in the image and a mask image of the complete scene. Let the mask of a single object or a hand be defined as $M_{\text{object only}}$ and that of complete scene as $M_{\text{scene}}$. The visibility ratio $VR$ of object $o$ is defined as:

$$VR_o = \frac{\sum_{i=1}^{N} I_o(M_{\text{scene}})}{\sum_{i=1}^{N} I_o(M_{\text{object only}})},$$

where $N$ is the number of pixels in a mask image, and $I_o(x)$ is the indicator function that returns one if pixel $x$ is an element of object $o$, and zero otherwise. If visibility
Table 2: Results on INTER-SH using Resnet50 as backbone. Left hand pose AUC results in % on our own dataset of synthetic closely interacting hands images categorized by the hand’s visibility ratio. Pose error in mm shown in parenthesis. AUC of 3D PCK is computed in an interval from 0 to 20 mm with 20 equally spaced thresholds. Bold faced and underlined text imply best and second-best score, respectively. The comparison is made against various methods that were either modified or recreated as code was unavailable.

| Model          | $VR$ value range |
|----------------|------------------|
|                | [0.40, 0.60]     | [0.60, 0.80] | [0.80, 0.95] | [0.95, 1.00] |
| Hason et al. [12]$^*$ | 0.43 (15.65)     | 0.47 (14.91) | 0.48 (14.60) | 0.49 (13.64) |
| Kulon et al. [21]$^*$ | 0.64 (10.24)     | **0.65 (9.79)** | 0.66 (9.58) | 0.71 (7.51) |
| Doosti et al. [7]$^*$ | 0.49 (13.86)     | 0.50 (13.78) | 0.50 (13.65) | 0.52 (12.56) |
| Lin et al. [24]$^*$ | 0.59 (11.27)     | 0.59 (11.59) | 0.60 (11.26) | 0.63 (9.67) |
| Ours           | **0.65 (9.69)** | 0.64 (10.28) | **0.66 (9.57)** | **0.71 (7.49)** |

Ratio is one, then the additional object should minimally affect the prediction capability of a hand-pose predictive model. In Sec. 5.3, we study the relation between the visibility ratio of hands in an image and the prediction score of various models for closely interacting hands.

5.3 Results

We report the pose estimation and mesh reconstruction accuracy of our model and compare it against other recent models. One of the models is the METRO model [24] that we have implemented ourselves, which produced state-of-the-art results on FreiHAND [52]. We also compared results to other models that produce mesh, keypoints, or both, namely [6], [28], [21], [7], and [12]. These last two models were originally designed to estimate both hand and object poses. For our experiments we modified them to predict positions of only hand mesh or joints. The modified methods are denoted as [12]$^*$ and [7]$^*$. Also, since the code model for [21] and [24] was not released by the authors, we have reproduced their models to the best of our availability and denote it as [21]$^+$ and [24]$^+$. 

Evaluation on FreiHAND. Table 1 compares the results of our model to others on the FreiHAND dataset. Our model outperforms previous works. In particular, we outperform the recent METRO model [24] that uses a deeper backbone than us. Since we also use a Transformer layer, like METRO, we further explore its contribution in our ablation studies below. Our model also improves upon [21] that uses a single path mesh decoding approach. This raises the question regarding the contribution of our multipath decoding strategy. This question is further explored in our ablation studies. We additionally compare the contribution of different image encoders to the mesh reconstruction task. Specifically, we experiment with Resnet50 [13], HR-Net with initial 18 and 48 channels [43]. Four output results using HR-Net 48 channels as backbone are shown in Figure 7(a).

Evaluation on INTER-SH. To further assess the performance of existing models on closely interacting hands, we train and test them using our generated hand-hand interacting images in INTER-SH. Since here we had to run and test other models, we either used a modified published original version, or a self-implemented version. Tab. 2 shows the models’ prediction evaluation metrics on the test set computed only for the left hand in the scenes. We additionally categorize the metrics based on the visibility ratio, defined in Section 5.2. Our method outperforms recently published algorithms. Four output examples of our model for this data are presented in Fig. 7(b). Further outputs are shown in the supplementary section.

It is not surprising to observe that all hand pose models improve their prediction accuracy when the visibility ratio of the hand increases. However, it is interesting to note that there is a relative large improvement at the $VR$ range of 0.95-1.00 compared to $VR$ range of 0.80-0.95. Thus, we learn that it is imperative to balance the dataset used for model training and testing to encompass different $VR$ ranges in order to obtain a more robust hand pose predictor. It is also interesting to notice that results on the
### Ablation studies

To investigate the contribution of the different components to the performance of our model, we conducted ablation studies in which we compare versions of our model with removed modules and stages. The versions were tested on the FrieHAND test dataset. The ablations contained removal of the attention block and removal of the multi-path component (reducing the model to a single path decoding). The results in Tab. 3 show that both the attention and multi-path components contribute to performance (notice that the attention block also adds depth). Specifically, comparing the single and multi path strategies that contain the exact same backbone and extract same network depth, we conclude that multi-path reconstruction is useful for the task of interacting hand mesh reconstruction.

### 6 Conclusions

We have proposed a new multi-stream graph convolution network that provides state-of-the-art results for hand mesh reconstruction from singular RGB image capturing two core scenarios: hand-object and hand-hand interactions. Scenarios were tested using both real and synthetic data. We additionally present a new method to generate hand-hand and hand-object interactions data that may benefit the hand pose estimation community.

### References

[1] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019. 1, 3

[2] 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 *The European Conference on Computer Vision (ECCV)*, August 2020. 1

---

**Table 3:** Results for ablation studies evaluated using online Freihand server. HRNet18 is used here as backbone for all experiments.

| Single path | Multi-path | Attention | PA-MPVPE | PA-MPJPE | F@5 mm | F@15 mm |
|-------------|------------|-----------|----------|----------|---------|---------|
| v           | v          | v         | 7.4      | 7.5      | 0.686   | 0.974   |
| v           | v          | v         | 7.3      | 7.4      | 0.69    | 0.975   |
| v           | v          | v         | 7.0      | 7.2      | 0.701   | 0.977   |

Figure 4: Mesh reconstruction results for our proposed model on the (a) FrieHAND dataset and (b) INTER-SH dataset. Left column is the original input and the right column is the predicted mesh.
[3] Yujun Cai, Liuhaoh Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision – ECCV 2018, pages 678–694, Cham, 2018. Springer International Publishing.  2

[4] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Hope-net: A graph-based model for hand-object pose estimation, 2020.  2, 6, 7

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2018.  3

[6] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In European Conference on Computer Vision, pages 769–787. Springer, 2020.  1, 2, 6, 7

[7] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David Crandall. Hope-net: A graph-based model for hand-object pose estimation, 2020.  2, 6, 7

[8] Guillermo Garcia-Hernando, Shaxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 409–419, 2018.  1

[9] Liuhaoh Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image, 2019.  1, 2

[10] J. C. Gower. Generalized procrustes analysis. Psychometrika, 1975.  6

[11] Shreyas Hampali, Markus Oberweger, Mahadi Rad, and Vincent Lepetit. HO-3D: A multi-user, multi-object dataset for joint 3d hand-object pose estimation. CoRR, abs/1907.01481, 2019.  1

[12] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatyk, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.  1, 2, 6, 7

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.  7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.  5

[15] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7):1325–1339, 2014.  6

[16] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression, 2018.  2

[17] J. Ketchel and P. Larochelle. Collision detection of cylindrical rigid bodies using line geometry. In Proceedings of IDETC/CIE, 2005.  12

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.  5

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR), 2017.  2

[20] Arno Knaptitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Trans. Graph., 36(4), 2017.  6

[21] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.  1, 2, 4, 6, 7

[22] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018.  4

[23] Fanqing Lin, Connor Wilhelm, and Tony Martinez. Two-hand global 3d pose estimation using monocular rgb. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 2373–2381, January 2021.  1, 3

[24] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. 2021.  1, 2, 6, 7

[25] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. IEEE Robotics Automation Magazine, 11(4):110–122, 2004.  2

[26] Gyeongsik Moon, Shouo i Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and base-
line for 3D interacting hand pose estimation from a single RGB image. 2020. 1, 2, 3

[27] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image, 2020. 6

[28] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In European Conference on Computer Vision (ECCV), 2020. 7

[29] Franziska Mueller, Florian Bernard, Oleksandr Sotynchenko, Srinath Sridhar, Rushant Mehta, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 49–59, 2018. 1, 2

[30] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotynchenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. ACM Transactions on Graphics (TOG), 38(4):1–13, 2019. 3

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In European conference on computer vision, pages 483–499. Springer, 2016. 2, 3

[32] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 1862–1869. IEEE, 2012. 3

[33] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 436–445, 2018. 1

[34] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, Computer Vision – ECCV 2018, pages 725–741, Cham, 2018. Springer International Publishing. 5

[35] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia), Nov. 2017. 2, 3, 4, 6

[36] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1145–1153, 2017. 1

[37] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints, 2020. 1, 2

[38] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation, 2018. 2

[39] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In European Conference on Computer Vision (ECCV), 2020. 1

[40] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 648–656, 2015. 3

[41] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision (IJCV), 2016. 1, 3

[42] 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 2, 4

[43] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 2020. 3, 4, 5, 7

[44] Guotian Xie, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. Interleaved structured sparse convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8847–8856, 2018. 3, 4

[45] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In International Conference on Learning Representations, 2019. 2

[46] L. Yang and A. Yao. Disentangling latent hands for image synthesis and pose estimation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9869–9878, 2019. 2

[47] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3D hand pose tracking and estimation using stereo matching. CoRR, 2016. 1
[48] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image, 2019.

[49] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):901–914, 2019.

[50] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multimodal data, 2020.

[51] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[52] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russel, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
7 Supplementary

In the following sections we explain in more detail our synthetic hand-hand and hand-object synthetic generator. In addition we present additional qualitative results of our proposed model on both Freihand and the Inter-SH dataset.

7.1 Mesh Movement via Local Coordinate Systems

Previous use of the MANO model includes randomizing the MANO parameters to generate random single hand pose examples, or using its mesh structure to generate examples of a hand grasping a static object. The grasp is generated by minimizing an energy function based on the hand-object distance. To generate close interactions of two dynamic meshes, we conceptually follow that latter method. The core part of our method is to define a local coordinate system (LCS) for each joint along a mesh’s kinematic chain (namely the index order of joints in the mesh).

**LCS for MANO model** is defined for each joint along the kinematic chain of the MANO hand model in its neutral pose. More formally, let the 16 joints of the hand plus 5 finger tips \( J^{3D} = [j_0^{3D}, \ldots, j^{3D}_{20}] \in \mathbb{R}^{21 \times 3} \), define a kinematic chain of the hand, where \( j_0^{3D} \) is the root joint. The kinematic chain starts at the wrist (root joint) and ends at the finger tips. For each non-root joint we establish the directions of the LCS as follows. First axis is defined as the direction of a joint to its immediate neighbour along the kinematic chain, \( \overrightarrow{z_i} = j_{i+1}^{3D} - j_i^{3D} \). To define the second axis \( \overrightarrow{x_i} \), we first locate a vertex \( v_i \) on the MANO mesh model such that the direction \( \overrightarrow{x_i} = v_i - j_i^{3D} \), approximately follows the flexion direction of the joint. \( \overrightarrow{x_i} \) is then the projection of \( \overrightarrow{x_i} \) onto the perpendicular plane to \( \overrightarrow{z_i} \). Finally, the remaining axis is computed as \( \overrightarrow{y_i} = \overrightarrow{z_i} \times \overrightarrow{x_i} \), where \( \times \) is the cross product. We normalize all axes vectors to a unit size.

**LCS for dynamic objects.** A similar process as performed on the MANO hand model to establish LCS for joints is done for dynamic mesh models, such as the chain model shown in Figure 2e-f.

7.2 Generating Mesh Interactions

With the mesh models defined in Sec. 7.1, we simulate interactions between two hand meshes, or a hand mesh and an object mesh using the following algorithm:

1. Place object or a hand at the center of the scene at a random pose and rotation.
2. Insert an additional hand model to the scene at a distance from the scene’s center such that the objects do not intersect. The added hand model is at a neutral pose such that the front of the hand is facing the object located at scene’s center.
3. Incrementally shift the hand model in (2) towards the scene’s center until a collision occurs between it and the object/hand at scene’s center.
4. Randomly move each joint of the dynamic models in the scene until collision with another object/hand occurs or maximum angle of rotation is reached for each joint. In our case, all angles of rotation for the hand models are limited to 60 degrees.
5. Repeat 4 till the dynamic models can no longer perform a valid move.
6. Render final scene

Figure 2 show an example of the workflow. The validity of hands’ location in 3D space is performed based on mesh intersection. Specifically, for fast collision check, we approximate the hand and static objects as a collection of cylinders and annuli. Then, we check whether a cylinder of one hand intersects a cylinder of another hand/object. To this end, we have implemented the algorithm of [17] that determines whether two cylinders intersect in a given configuration. We check in a similar manner the possible intersection of fingers located on the same hand given a hand pose.

7.3 Additional qualitative results

We present qualitative results on Freihand and Inter-SH in Figs 5, 6, 7, 8.
Figure 5: Mesh reconstruction results for our proposed model on the INTER-SH dataset, Hand-Hand interaction images. Right column in the original input, central column shows projected 2D keypoints of the estimated 3D model on top of the original input frame, and the third column in the predicted 3D mesh.
Figure 6: Similar to Fig. 5 only focusing on Hand-Chain interaction images.
Figure 7: Mesh reconstruction results for our proposed model on the FrieHAND dataset.

Figure 8: Similar to Fig 5 only showing incorrect predictions for Hand-Hand interaction images.