Hand Pose Understanding with Large-Scale Photo-Realistic Rendering Dataset

Xiaoming Deng, Yinda Zhang, Jian Shi, Yuying Zhu, Dachuan Cheng, Dexin Zuo, Zhaopeng Cui, Ping Tan, Liang Chang, Hongan Wang

***Abstract*—Hand pose understanding is essential to applications such as human computer interaction and augmented reality. Recently, deep learning based methods achieve great progress in this problem. However, the lack of high-quality and large-scale dataset prevents the further improvement of hand pose related tasks such as 2D/3D hand pose from color and depth from color. In this paper, we develop a large-scale and high-quality synthetic dataset, PBRHand. The dataset contains millions of photo-realistic rendered hand images and various ground truths including pose, semantic segmentation, and depth. Based on the dataset, we firstly investigate the effect of rendering methods and used databases on the performance of three hand pose related tasks: 2D/3D hand pose from color, depth from color and 3D hand pose from depth. This study provides insights that photo-realistic rendering dataset is worthy of synthesizing and shows that our new dataset can improve the performance of the state-of-the-art on these tasks. This synthetic data also enables us to explore multi-task learning, while it is expensive to have all the ground truth available on real data. Evaluations show that our approach can achieve state-of-the-art or competitive performance on several public datasets.**

***Index Terms*—Hand pose estimation, Photo-realistic synthetic dataset, Physical-based rendering, Multi-task CNN.**

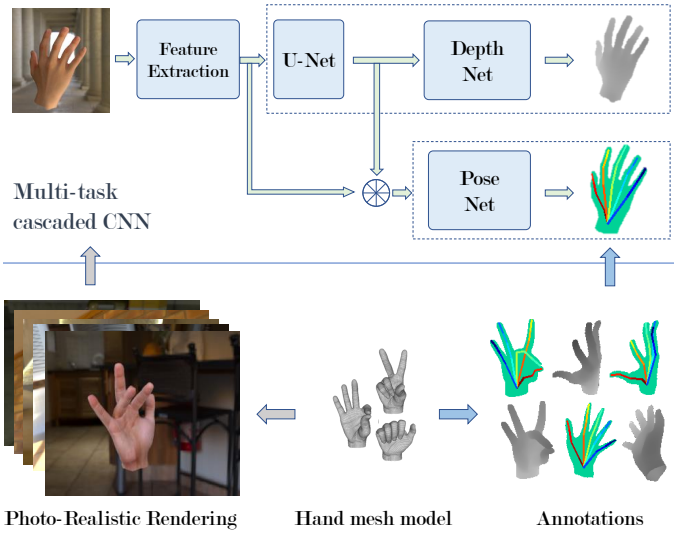


Fig. 1. The overview of our work. We build a large-scale photo-realistic hand pose database with a variety of hand shapes and poses, and render millions of synthetic images with different kinds of ground truths. Our dataset with multiple annotations enable us to design a multi-task CNN structure that predict depth and 3D hand pose from input RGB image in a cascaded way.

Manuscript received July 18, 2019; revised February 29, 2020, and October 20, 2020; accepted March 15, 2021. Date of publication XX XX, 2020; date of current version XX XX, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2019YFC1521100, and in part by the Distinguished Young Researcher Program, Institute of Software, Chinese Academy of Sciences. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Senem Velipasalar. (*Xiaoming Deng, Yinda Zhang and Jian Shi contributed equally to this work*) (*Corresponding author: Xiaoming Deng*)

X. Deng, Y. Zhu, D. Zuo, H. Wang are with the Beijing Key Laboratory of Human-Computer Interactions, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. E-mail: xiaoming@iscas.ac.cn, hongan@iscas.ac.cn

Y. Zhang is with Google, 1600 Amphitheatre Pkwy, Mountain View, CA, USA, 94043. (e-mail: yindaz@google.com)

J. Shi is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. (e-mail: jian.shi@ia.ac.cn)

D. Cheng is with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China.

Z. Cui is with the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China. (e-mail: zhpcui@gmail.com)

P. Tan is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada, and also with Alibaba, Hangzhou, China. (e-mail: pingtan@sfu.ca)

L. Chang is with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China. (e-mail: changliang@bnu.edu.cn)

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material contains additional comparison experiments with other methods and ablation study. Contact xiaoming@iscas.ac.cn for further questions about this work.

I. Introduction

As an important task in computer vision, recovering hand pose is essential for many applications in the area of human-computer interaction, augmented reality, and robotics [1]. One type of the most popular approaches to address this problem are deep learning based methods, which learn the hand pose from large-scale dataset. One key bottleneck of deep learning based hand pose methods is the lack of large-scale hand pose dataset with diverse and accurate annotations. Existing datasets for hand pose understanding are very limited in terms of number of frames, number of subjects, level of annotations and annotation accuracy. For example, BigHand2.2M [2] is the largest public available hand pose dataset, which only contains depth and 3D joint annotations. In order to collect BigHand2.2M, the subjects have to wear a marker-based tracker on the hand to get ground truth. As a result, the color image could be contaminated and can not be applied to color-based hand pose estimation task. RHD [3] contains diverse annotations, but it only has about 44K images. Moreover, RHD is not photo-realistic and has a great domain gap to real color images. Collecting large-scale hand pose dataset is not trivial, and it needs accurate 3D hand pose estimation as well as labor-intensive efforts for semantic segmentation.

To reduce the cost for building real datasets, synthetic data has been widely employed in recent works [3], [4], [5], [6]. Compared to real data, synthetic data is easy to scale up and provides high-quality ground truth. Another advantage of synthetic data is the availability of diverse supervisions. For example, besides the 3D hand pose ground truth, many other kinds of supervision can be obtained from virtual hand models, e.g. depth and semantic segmentation, which could be potentially useful for pose estimation. One of the major concerns about the synthetic data is the domain gap on the input side, e.g. color, depth, between the rendered synthetic data and real data. The other problem is how hand pose understanding related tasks are affected by large-scale datasets.

To address these problems, we develop a large-scale synthetic dataset (about 5.52M images), namely PBRHand. Images are rendered from textured hand meshes under a variety of hand poses. To ensure the reality, we capture high-fidelity 3D hand models and pose sequence with professional 3D scanners and digital glove. We randomly sample a variety of camera viewpoints and environment map illuminations, and employ physically based rendering (PBR) to generate photo-realistic images. As for the ground truth, we generate accurate 3D skeleton hand pose, pixel-wise semantic segmentation (e.g. finger, palm) and depth maps.

We use our data to pretrain convolutional neural network for three hand pose understanding related tasks, 2D/3D hand pose from color, depth estimation from color image, and 3D hand pose from depth. The experiments demonstrate that for these hand pose related tasks our data can improve the network performance of the current state-of-the-art. We also show that for 3D pose from single color image the dataset enables us to pretrain our model with multiple supervisions and delivers better performance after finetuning on real datasets (see Fig. 1). Specifically, we present a cascaded multi-task deep neural network based on PBRHand for hand pose estimation from color image, which can effectively learn from an auxiliary depth recovery task.

Compared to Weakly-sup [7] that also leverages depth map for auxiliary supervision during the hand pose estimation, our cascaded multi-task method adopts a different way to use depth map. Inspired by the success of hand pose from depth, we consider depth map/feature can be used as important input for hand pose, and we conduct an in-depth analysis on the effect of different cascaded forms (feature cascade, input/output cascade), and find that the depth feature can improve the 3D hand pose estimation from color (See Section VI.B). Weakly-sup [7] presents a weakly-supervised network for 3D hand pose estimation. They predict 3D hand pose from the input color image, and then generate depth map from the predicted 3D hand pose. The generated depth map serves as weak supervision for 3D pose regression. Moreover, as shown in Table II of our supplementary document, our method achieves better performance than [7].

The major contributions of this work are as follows:

1) We develop a high-quality synthetic hand pose dataset, namely PBRHand. It contains a large number of photo-realistic synthetic color images with various ground truths (depth maps, semantic segmentation labels and hand joints);
2) We demonstrate that the networks pretrained on our PBRHand achieve state-of-the-art performances on three hand pose related tasks. We also shows that the rendering quality is important for hand pose estimation;
3) Our PBRHand also enables us to explore multi-task learning. We present a cascaded multi-task deep neural network based on PBRHand for hand pose estimation from color image. Our network can effectively learn from an auxiliary depth recovery task for 3D hand pose estimation. The network achieves the state-of-the-art or competitive performance on public datasets.

The rest of the paper is organized as follows. In Section II, we introduce related works. In Section III, we introduce the pipeline of generating synthetic hand pose dataset. In Section IV , we investigate the effect of our dataset on three typical hand pose related task. In Section V, we present our multi-task model for hand pose estimation in detail. In Section VI, we show comparison experiments and ablation study. Conclusions are given in Section VII.

II. Related Work

Building large-scale dataset is a key issue for hand pose estimation using convolutional neural networks.

One direction is to construct real hand pose datasets, in which the images are captured with a real color or depth camera, and the joints are annotated manually, by model fitting methods [16], or jointly using both methods [17]. Constructing large-scale dataset with manual annotation is almost prohibited, due to the fact that it usually requires excessive human efforts, results in inaccurate annotations, and it is also time-consuming for scaling up. ICVL, NYU and MSRA dataset are three main depth image benchmark datasets for hand pose performance evaluations. ICVL dataset [17] adopts human pose tracking and manual refinement for annotation. However, the total numbers of frames and subjects are small, and its issue of annotation accuracy is found in [18]. MSRA dataset [19] uses the same strategy for annotation. Its scale is larger, but the annotation accuracy is still not high. NYU dataset [16] is a larger benchmark with a large view coverage, large variation in articulation. It relies on first model-based fitting, and then particle swarm optimization to get the joint annotation. Manually adjusting annotation and re-initialization are required. Oberweger *et al.*[18] also propose a semi-automated method for labeling a hand depth video with the 3D joints. The dataset currently contains only 4 subjects and about 63K RGBD frames. Recently, Yuan *et al.* propose BigHand2.2M [2]. BigHand2.2M is currently the largest real dataset for depth, and the hand joints are annotated with magnetic markers. The dataset is of less diversity with respect to hand shape with only 10 subjects. Moreover, it is not suitable for color-based hand pose estimation tasks. Datasets such as Stereo [20], Dexter+Object [10], Dexter Ego [4], MPII+NZSL [21] for color based hand pose estimation are usually either relatively small or contain only partial annotations. Recently, FreiHands [9] is created as a large-scale real dataset of color images with hand pose/shape labels;

TABLE I
DATASET COMPARISON. 'PARTSEG' IS ABBREVIATION FOR HAND PART SEGMENTATION. 'FINGERTIPS' MEANS 5 FINGERTIPS ARE LABELED, AND IT CAN BE EXTRACTED FROM 'JOINTS'. OUR DATASET HAS BETTER DATA VARIATIONS, QUALITY OF RENDERED IMAGES AND FULL ANNOTATIONS.

Dataset	Content	Type	Frames	Subjects	Viewpoints
Stereo [8]	RGB, depth, joints	real	18K	6	3rd
FreiHands [9]	RGB, joints	real	170K	-	3rd
Dexter+Object [10]	RGB, depth, fingertips	real	3,014	2	3rd
Dexter Ego [4]	RGB, depth, fingertips	real	3,190	4	ego
FingerPaint [11]	depth, fingertips	synthetic	100k	1	full
HandNet [12]	depth, 6 joints	real	212k	10	3rd
MHP [13]	depth, joints	real	80k	9	3rd
Bighand2.2M [2]	depth, joints	real	2.2M	10	full
RHD [3]	RGB, depth, joints, partseg	Blender [14]	44K	20	full
Our PBRHand	RGB, depth, joints, partseg	Mitsuba [15]	5.52M	50	full

However, the background is synthetic, and the annotation is semi-automatic and requires human-in-the-loop verification.

The other direction is to build synthetic dataset, which has shown promising effect for convolutional neural network training of many computer vision tasks, indoor scene understanding [5], intrinsic image [6], optical flow [22][23], object pose estimation [24], and human pose estimation [25]. These methods employ synthetic dataset with ground-truth annotations for network pretraining, and then finetune on real dataset. Synthetic data for human hand is especially challenging due to the fact that it is generally hard to capture high-quality hand shapes and hand motions from real subjects as well as generate photo-realistic images to narrow the gap between the real images and synthetic images. Zimmermann and Brox [3] used a rendered hand dataset named RHD with various ground truth for pose estimation. The dataset is almost limited with the number of frames (44K) and the number of human subjects (20). In contrast, we provide 125 times of images in RHD, and our rendered color images are more photo-realistic than the color images in RHD. SynthHands [4] contains depth and color images from an egocentric viewpoint, and the images are rendered using Unity. The dataset is the first synthetic dataset with hand interaction with objects. GANeratedDataset [26] translates synthetic images in SynthHands to real images using geometrically consistent CycleGAN. The dataset contains more than 330K color images of hands with 2D and 3D hand pose annotation. The MSRC FingerPaint hand pose dataset [11] contains 100K synthetic depth frames for one subject, and its hand poses are limited by random sampling from six articulations [2].

Our dataset proposed in this paper differs from previous datasets. It is the first large-scale hand dataset with photo-realistic RGB, depth, semantic segmentation, and 3D hand joints. The whole data generation pipeline is of little cost and can be scaled up easily. We show that our data with shape, pose, texture and illumination variations can benefit hand pose understanding problems.

III. PBRHAND HAND POSE DATASET

We build a large-scale hand pose dataset, named PBRHand (physically based rendered hand). It contains millions of photo-realistic color images and various kinds of accurate ground truths which could be potentially helpful for hand pose related tasks. Recently, deep learning methods have been successfully applied to hand pose estimation problem, and several good datasets are publicly available (Table I). Those datasets made great contributions to improve the quality of this task. However, there are still some limitations. First, for real data, it is extremely expensive to capture and label large-scale datasets for training complex deep neural networks. Current real datasets [20], [4] are only built with thousands of frames and quite few subjects, which limits the generalization of deep neural networks. Recently, Zimmermann and Brox [3] build a synthetic dataset named rendered Hand Pose Dataset (RHD), which contains more frames, poses and subjects. The data variation of this synthetic dataset is generally better than existing real datasets. However, since they use low-quality 3D human models and unrealistic rendering method, there exists a large gap between the dataset and real-world images.

To bring further improvement to hand pose understanding, we choose to build a new dataset. To make the data realistic and diverse, we first use professional devices to capture hand models and poses from human subjects. We manually clean and label those scanned 3D hand models. After that, we apply photo-realistic rendering to generate millions of images with multiple ground truths. Overall, our PBRHand contains 5.52 million synthetic color images with 3D hand pose, semantic segmentation and depth. Fig. 2 shows several examples of our photo-realistic hand dataset. Table I shows the comparison of our dataset and other datasets.

A. 3D Hand Model and Pose

To collect high-fidelity mesh models, we use a hand-held 3D scanner Artec ® Eva scanner [27] to capture dense 3D point cloud of hands in standard upright pose (Fig. 3 (a)). For the diversity of hand shapes between human, we collect hand shapes from 50 people, including 30 males and 20 females of different ages, careers and races. We scan a human hand at rest pose by moving Eva scanner around toward the hand, soften point cloud and remove noise of each frame using 'Smooth Brush' and 'Eraser' tools in Eva capture software, then fuse multiple scans into a completed mesh using the 'Fusion' tool. Though Artec Eva scanner can provide color texture for a captured mesh, we found the texture was blurry

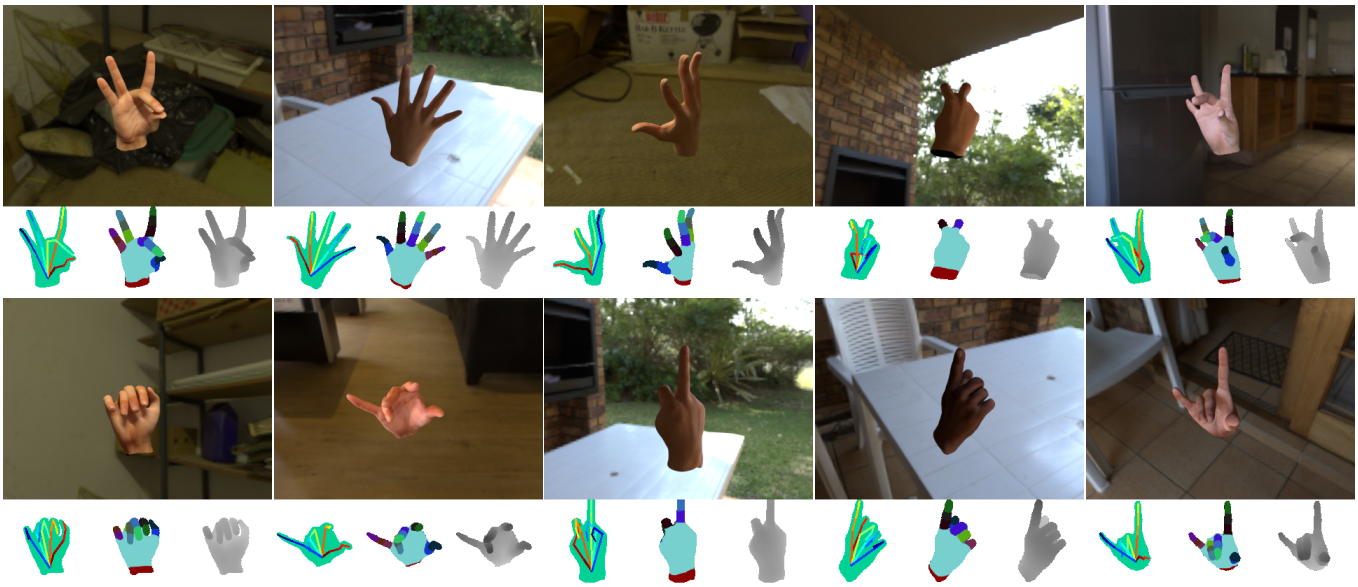


Fig. 2. Our photo-realistic hand dataset by posing real hand models with real hand motion data. Examples of rendered photo-realistic hand color images are shown on the top rows. Examples of hand poses, rendered depths and semantic segmentation are shown on the bottom rows.

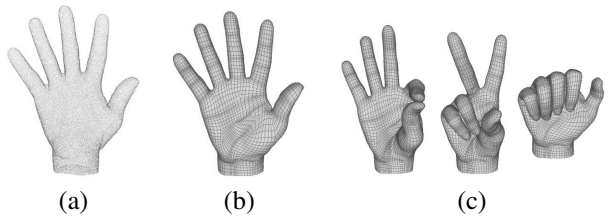


Fig. 3. Hand model generation. We show captured hand point cloud with 3D scanner in (a), rewired mesh in (b), and mesh deformation under several poses in (c). We use real hand mesh models instead of hand meshes designed by artists, which are broadly used in previous hand pose estimation literature.

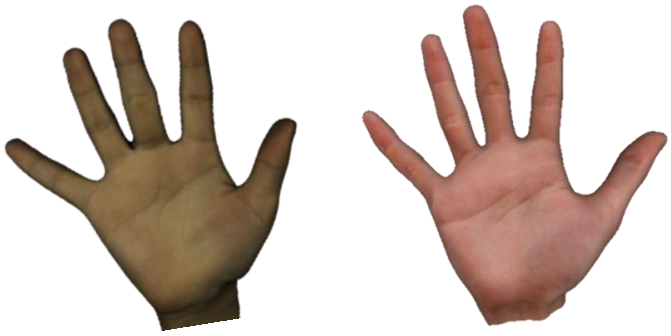


Fig. 4. Textured hand shape. left: original texture; right: improved by designer.

in some areas and there were also some artifacts. Therefore, we employ professional designers to manually refine the hand textures for high reality and rich details (Fig. 4). In order to simplify the rigging and texture mapping process, we rewire the surface line to fit natural hand topology of a standard hand mesh template using nonrigid ICP [28] (Fig. 3 (b)). In our current hand model, we do not add arm part. Adding arm while still maintaining the reality is a challenging task. Capturing 3D models for arm and dealing with skin, pose, and potential interaction with clothes requires huge effort. This is a great direction for future work. The point clouds are then fitted to triangulated textured mesh using UV mapping [29]. The texture maps of our hand models are manually created by professional engineers according real hand texture images.

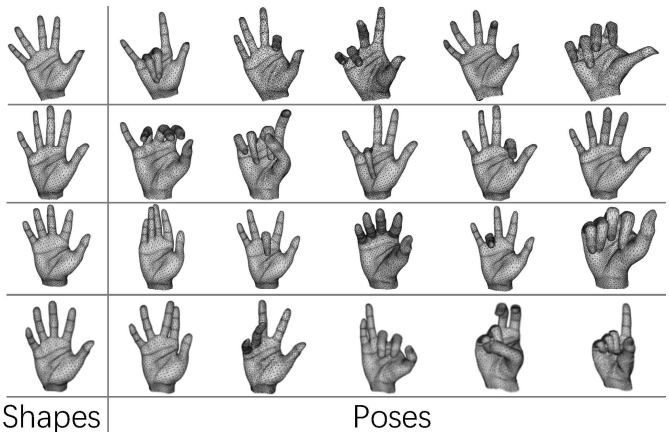


Fig. 5. Hand shapes and poses. With hand models of 50 subjects and $1,840$ gestures, we synthesize 92K hand meshes with various hand shapes and poses.

To capture realistic hand poses, we summarize $1,840$ key hand poses for human-computer interactions inspired by existing datasets such as NYU [16], Dexter+Object [10] and EgoDexter [4]. We ask subjects to show these poses wearing a Measurand® ShapeHand glove, which captures skeleton poses. The key pose set covers typical hand gestures such as grasping, pinch, abduction-abduction of all fingers together, flexion-extension of all fingers etc. The subjects are also asked to move hand freely to further increase the pose diversity. The captured hand poses usually contain noises, so we employ professional designers to refine local joint position and orientation. It makes hand poses right and reduces artifacts.

The collected hand poses are refined and semi-automatically

Fig. 6. HDR environment maps used for rendering our photo-realistic hand images. They consist of 10 indoor scenes and 10 outdoor scenes under different lighting conditions.

re-targeted onto the collected hand mesh models using rigging and deformation [30], so that the hand meshes can be transformed automatically into the collected pose set. Fig. 3 and Fig. 5 shows some examples of hand mesh deformation under several poses. The captured hand poses usually contain noises, so we employ professional designers to refine local joint position and orientation. It can reduce hand mesh artifacts such as mesh penetrations during the model re-targeting. We first manually align the standard hand pose to hand meshes (only one mesh per subject) by adjusting joint local offsets. And then we bind the skeleton and mesh in Autodesk ® Maya, and we can generate meshes automatically under different poses by deformation. However, the deformation may have distortions to the smooth binded skin, we employ professional artists to manually check the data, adjust the skin weights and control the deformation, and make sure that the rigged hand meshes are of high quality.

Moreover, with the current data collection framework, we are able to capture high quality pose and shape and create photo-realistic texture within a reasonable amount of time. Firstly, we collect a hand pose set of 1,840 poses using a Measurand® ShapeHand glove within less than two hours. Secondly, we collect hand shapes, create hand textures, and retarget the hand poses for 50 subjects. The manual work for each subject costs about 85 minutes. Table II shows the detailed time of manual work for each subject.

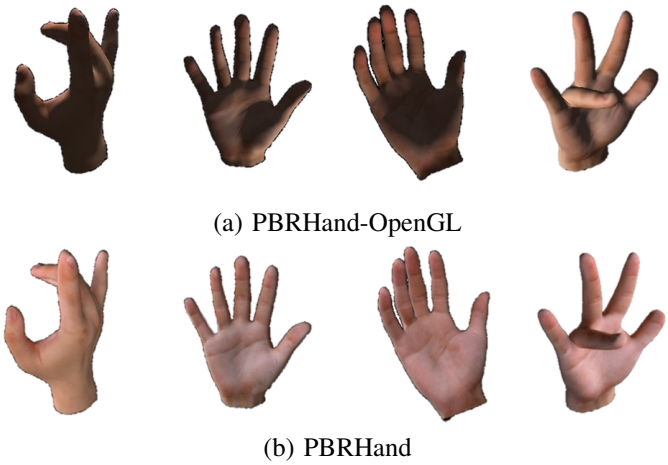
TABLE II
MANUAL WORK (IN MINUTE) OF HAND SHAPE CAPTURE AND TEXTURE CREATION FOR EACH SUBJECT.

	Capture hand shape and post-process	Create hand texture	Rig and deform hand shape with pose
Time	10	60	15

B. Physically Based Rendering

With high-fidelity hand meshes and realistic pose, we employ a widely used physically-based renderer Mitsuba [15] to generate photo-realistic images. We pick camera viewpoints with random distances in $[400, 800]$ mm away from the center of the mass of the hand model, and sample 3 camera positions with the zenith angle in $[\frac{1}{6}\pi, \frac{2}{3}\pi]$ and the azimuth angle in $[0, 2\pi)$. To simulate realistic illumination effect, we collect 20 HDR environment maps, where indoor and outdoor scenes with various illumination conditions are included. Fig. 6 shows these environment maps. The perspective camera has a resolution of 640×480, horizontal field of view of 54 degrees. Given a camera position, our hand models are rendered under a random subset of the environment maps using path tracing. With the hand meshes, textures, environment maps and random sampled viewpoints, we obtain more than 5.52 millions hand images by photo-realistic rendering. Fig. 2 illustrates the synthetic image, depth, segmentation and joints of our dataset.

In order to compare whether photo-realistic rendering is helpful for hand pose understanding tasks, we also render from unrealistic rendering with directional lights using OpenGL. Our dataset rendered with OpenGL is named PBRHand-OpenGL. Basic shading effects, multiple directional lights are included, but no global illumination, or shadow is included. Fig. 7 shows examples with OpenGL and physically-based rendering. We only show the hand regions to better illustrate the effect of different rendering methods on hand regions.



(a) PBRHand-OpenGL

(b) PBRHand

Fig. 7. Hand image rendered with different methods. (a) OpenGL. (b) Physically-based rendering.

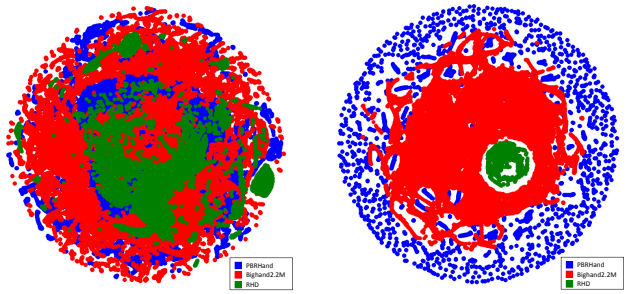
C. Annotation

For each sample in our PBRHand dataset, we provide rendering color images with semantic segmentation, depth image and 3D joint coordinates. When modeling hand shape, designers manually label 15 hand parts' segmentation on the hand meshes of a standard pose, the segmentation is stored as a texture map, and the segmentation of hand mesh undergoing different poses can be automatically labeled during the hand pose retargeting process. The semantic segmentation is also generated by rendering the albedo of the hand mesh with semantic label encoded texture map. The depth is extracted from z-value of 3D position in camera space. The depth is rendered using "distance field" integrator followed by a post-processing that converts the ray distance to camera center into depth with camera intrinsic matrix. When modeling hand poses, hand meshes are deformed according to the pose of its

corresponding skeleton. We calculate the joint locations in the camera space with the transformation and forward kinematics.

D. Hand Pose/Shape Space

Following Bighand2.2M [2], we visualize the hand pose/shape space, and compare our PBRHand dataset with the existing datasets using 2D t-SNE. Fig. 8 shows the projected hand pose space and hand shape space of our PBRHand dataset, RHD [3] and BigHand2.2M [2]. We observe that our dataset has better coverage in hand shape space, and the hand pose distribution of our dataset and Bighand2.2M is better than RHD. It is consistent with the intuition that the pose space is simpler than the shape space, where the hand shape can be more difficult to be captured and the hand shape is different between subjects.



(a) Hand pose space (b) Hand shape space

Fig. 8. 2D t-SNE of hand pose and hand shape space for PBRHand, BigHand2.2 and RHD dataset.

IV. Hand Pose Understanding Related Tasks

In this section, we first investigate three fundamental hand pose related tasks: (1) 2D/3D hand pose estimation from color image; (2) depth estimation from color image; (3) 3D pose from depth image. For these tasks, we show how the state-of-the-art model pretrained with our synthetic dataset compares with the model pretrained with other synthetic dataset.

A. Dataset and Evaluation Metrics

In this paper, we evaluate hand pose estimation performance for different hand pose understanding tasks on several public datasets: Stereo dataset [8], MPII-NZSL dataset [21], NYU dataset [16], Dexter+Object dataset [10], Rendered Hand Pose dataset (RHD) [3], and Task3 of Hands 2019 Challenge (Hands-2019) [31]. In this section, we use Stereo dataset, MPII-NZSL dataset and NYU dataset for evaluation. In Section V, we use Stereo, RHD, Dexter+Object and Hands 2019 datasets for evaluation. We briefly review them in the following and present the evaluation metrics.

a) Stereo Hand Pose dataset (Stereo) [8]: is the largest public real image dataset with fully annotated 3D hand joints using color images. We use the color-depth subset of Stereo SK captured from a RGBD camera. It provides color and depth images, 3D annotations for 21 hand keypoints. The dataset is separated into an evaluation set of $3,000$ images and a training set of $15,000$ images.

b) MPII+NZSL dataset [21]: contains 2800 images with 2D hand pose annotations. There are about 2000 and 800 images for training and testing.

c) NYU hand dataset [16]: is a depth image based hand pose dataset. We use NYU dataset to compare the effect of our scanned hand shape and the parameteric MANO model on 3D hand pose estimation from depth. However, we do not evaluate on the color images of NYU dataset, because these images are registered with depth and thus corrupted [3].

d) Rendered Hand Pose dataset (RHD) [3]: is a synthetic dataset, which consists of 41,258 images for training and 2,728 images for evaluation. The images are rendered using Blender. The dataset provides 2D and 3D joint annotations, depth map, and part segmentation.

e) Dexter+Object dataset [10]: consists of 6 sequences with 2 actors, and varying interactions with a simple object shape. Fingertip positions and cuboid corners were manually annotated for all sequences. Hand pose estimation on Dexter+Object Dataset is very challenging, since no training dataset is available and there are heavy occlusions due to the interaction between hand and object.

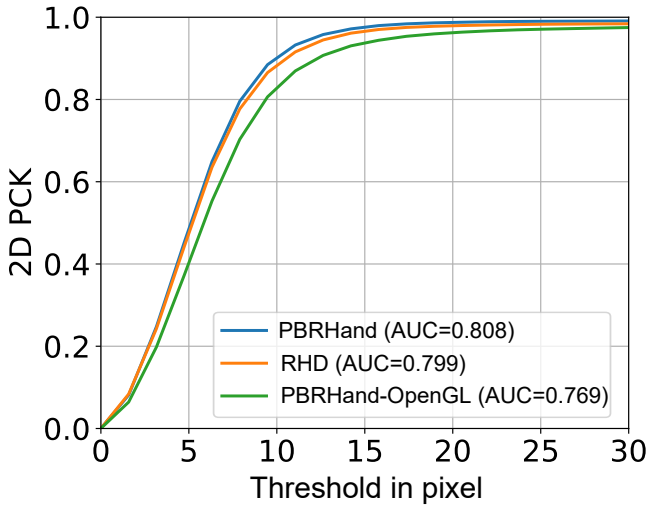
f) Task3 of Hands 2019 Challenge dataset (Hands-2019) [31]: contains images from 3 subjects manipulating 4 objects for training, and contains images from 5 different subjects manipulating 6 different objects for testing. Hands-2019 dataset is also challenging due to the heavy occlusion of objects.

Evaluation Metrics We conduct three hand pose understanding related tasks, and for each task we adopt the standard evaluation metrics in literature. To evaluate 2D and 3D hand pose accuracy from color image, we follow common metrics [3] and use average End-Point-Error (EPE), Area Under the Curve (AUC) and percentage of correct keypoints (PCK) over different thresholds.

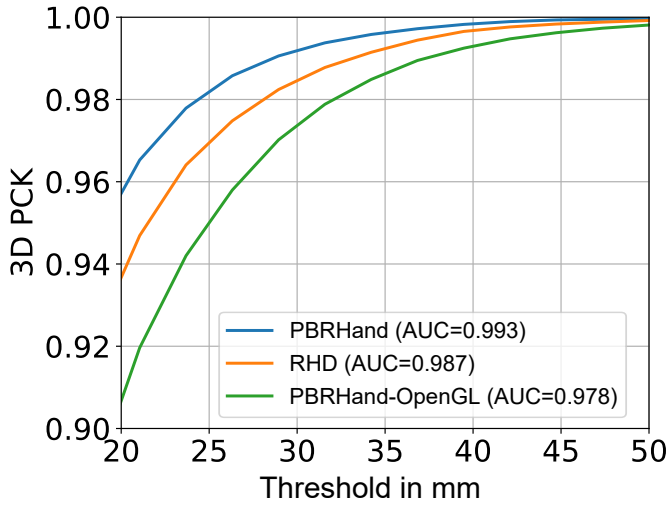
For 3D hand pose from color image, we follow [3] to estimate normalized 3D joint coordinates relative to a reference point, which can be selected as a hand root joint or the 3D center-of-mass (CoM) of hand foreground region estimated from the aligned depth of the color image. In the evaluation of Stereo, RHD and Dexter+Object, we use 3D CoM as the reference point. In the evaluation of Hands-2019, we use the provided ground-truth wrist locations for the test images as the reference point. During the evaluation, the absolute joint positions are obtained by adding the predicted normalized 3D joints with the reference point. For MPII+NZSL dataset, we follow [21][32] to use handsize-normalized PCK for evaluation. For depth estimation from color, we use mean absolute error (MAE) to evaluate the difference between the ground truth and estimated depths on the hand foreground. To evaluate 3D hand pose from depth, we use the standard metrics proposed in [17], the mean joint errors and the percentage of test images that have all predicted joint errors within a given distance threshold from the ground truth.

B. 2D/3D Pose from Color image

Method. We use the network architecture in CHP3D [3] for 2D/3D hand pose estimation from a single color image. The network consists of three blocks, HandSegNet, PoseNet and



(a) 2D hand pose results with PoseNet.



(b) 3D hand pose results with PoseNet + PosePrior.

Fig. 9. (a) 2D hand pose results with PoseNet. (b) 3D hand pose results with PoseNet + PosePrior. PoseNet and PoseNet + PosePrior are both pretrained on different datasets and finetuned on Stereo dataset.

PosePrior network. HandSegNet is a segmentation network to segment hand mask. With the hand mask predicted by HandSegNet, the input image is cropped and normalized in size, and fed to PoseNet to get the score maps of 2D hand joints. The PoseNet adopts a cascaded encoder-decoder network similar to [33] to predict score map of the input image. PosePrior recovers 3D hand joints using the score map of 2D hand joint as input.

Training. Similar to CHP3D [3], we first initialize the first 16 layers of PoseNet network with pretrained weights of [33] and all the other layers with random weights, and train the PoseNet on the Stereo dataset [8] and the synthetic dataset. Following [3], we use the score map predicted by PoseNet to train PosePrior. We keep PoseNet fixed and train on Stereo hand pose dataset (Stereo) [8]. To highlight the difference between physical-based rendering and unrealistic OpenGL rendering techniques, we train on PBRHand + Stereo, PBRHand-OpenGL + Stereo and RHD [3] +Stereo. The networks are trained with a batch size of 16 and using ADAM solver. The learning rate is 0.0001 for the first 10K iteration, 0.00001 for the following 10K iterations and 0.000001 until the end.

Experiment. We train with the code released by [3] and test on the Stereo dataset [8]. Firstly, we compare the performance of 2D keypoint estimation using PoseNet trained on our PBRHand dataset, PBRHand-OpenGL dataset and the RHD dataset. Secondly, we compare the performance of 3D joints using PoseNet + PosePrior. Fig. 9(a) and Table III show the performance of 2D pose estimation on the Stereo dataset with different training settings. Fig. 9(b) and Table IV show the performance of 3D pose from color image on Stereo dataset with different training settings. We observe that the EPE and AUC (20-50mm) with the 3D pose model pretrained on our PBRHand dataset are better than the model without pretrain, the models pretrained on RHD dataset [3] and PBRHand-OpenGL dataset. Moreover, the 2D hand pose with the model pretrained on PBRHand is also better than the model pretrained on RHD and PBRHand-OpenGL. Fig. 10 shows 2D hand pose results on Stereo dataset with different training datasets. We can see that the 2D hand pose with the model pretrained on PBRHand provides reasonable hand skeleton joints and aligns better with the hand foreground than those with the models pretrained on RHD and PBRHand-OpenGL.

We also test the effect of our PBRHand dataset on MPII+NZSL hand pose dataset [21]. MPII+NZSL only contains 2D hand pose annotations, and we evaluate the effect of different synthetic datasets on 2D hand pose using PoseNet. Fig. 11 and Table V show the performance of 2D pose estimation on the MPII+NZSL dataset with the PoseNet models trained on different datasets. We observe that the 2D hand

TABLE III
PERFORMANCE OF 2D HAND POSE FROM COLOR IMAGE ON STEREO DATASET WITH DIFFERENT TRAINING DATASETS. '-' IN THE FIRST COLUMN MEANS NO PRETRAINING. AUC IS SHOWN IN PERCENTAGE POINTS.

Pre-train Dataset	PoseNet	
	AUC(0-30pix.)↑	EPE (pix.) ↓
-	76.1	8.732
RHD	79.9	7.500
PBRHand	**80.8**	**6.467**
PBRHand-OpenGL	76.9	8.203

TABLE IV
PERFORMANCE OF 3D POSE REGRESSION FROM COLOR IMAGE ON STEREO DATASET WITH DIFFERENT TRAINING DATASETS. '-' IN THE FIRST COLUMN MEANS NO PRETRAINING. AUC IS SHOWN IN PERCENTAGE POINTS.

Pre-train Dataset	PoseNet + PosePrior	
	AUC(20-50mm) ↑	EPE (mm) ↓
-	97.9	9.142
RHD	98.7	8.626
PBRHand	**99.3**	**7.716**
PBRHand-OpenGL	97.8	9.502

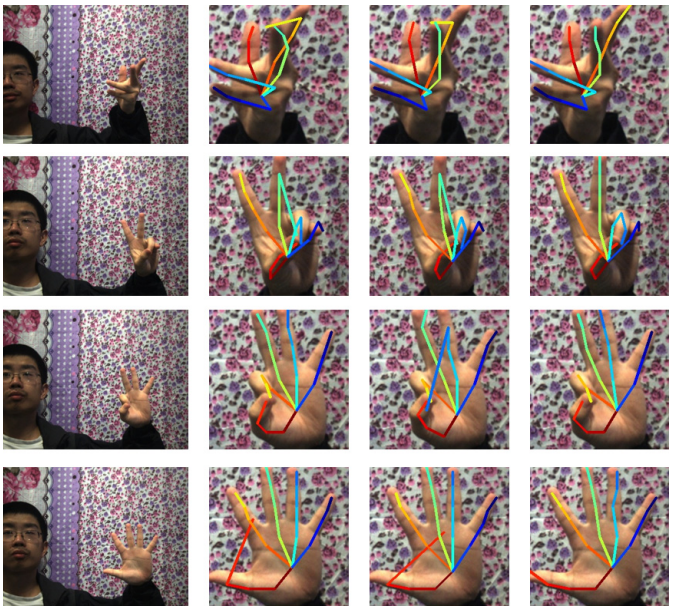


Fig. 10. Hand pose results on Stereo dataset with different training datasets. The pretrained model on PBRHand provides more reasonable hand skeleton joint structure.

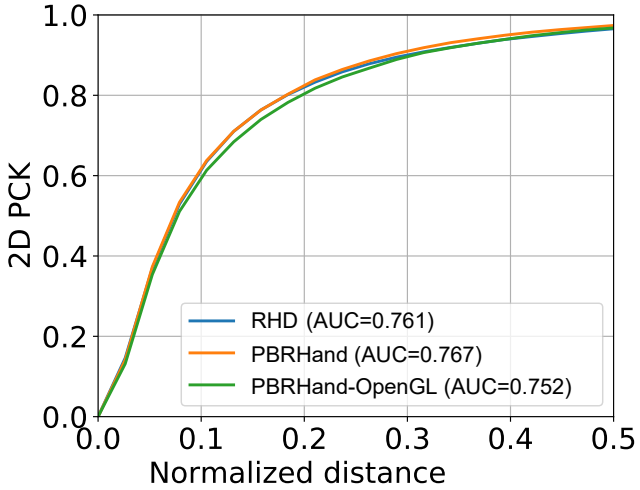


Fig. 11. 2D hand pose results with PoseNet pretrained on different datasets and finetuned on MPII+NZSL dataset.

pose with the model trained on PBRHand is better than those with the model without pretraining, the models pretrained on RHD dataset and PBRHand-OpenGL dataset.

Therefore, our PBRHand dataset synthesized by physically-based rendering improves the performance of color-based hand pose estimation upon the state-of-the-art method.

C. Depth Estimation from Color Image

Method. We utilize a stacked hourglass network to regress depth estimation from color image similar to human depth estimation task in [34]. The network formulates the depth estimation from a color image as a pixel-wise regression problem. Following the image preprocessing in [34], we first normalize the depth values of hand foreground pixels to $[-1, 1]$. We find the 3D center of the mass (CoM) of the hand region using the depth map, then the depth map is normalized by subtracting CoM and divided by a constant 100. The network adopts convolutional layers with residual connections and 3 stacked 'hourglass' modules, and each followed module takes the prediction of the previous module as input.

TABLE V
PERFORMANCE OF 2D HAND POSE FROM COLOR IMAGE ON MPII+NZSL DATASET USING POSENET TRAINED UNDER DIFFERENT TRAINING DATASETS. '-' IN THE FIRST COLUMN MEANS NO PRETRAINING. AUC IS SHOWN IN PERCENTAGE POINTS.

Pre-train Dataset	PoseNet	
	AUC(0-0.5 hand size)↑	EPE (pix.) ↓
-	65.5	12.197
RHD	76.1	12.197
PBRHand	**76.7**	**11.803**
PBRHand-OpenGL	75.2	12.708

Training. The training process consists of two stages, pretraining and finetuning. We pretrain the model on the synthetic datasets (PBRHand and RHD), then finetune the pretrained model on the Stereo pose dataset [8] similar to Zhang *et al.*[5]. We use RMSProp optimizer to train and finetune the model. At the pretraing stage, the initial learning rate is set to 5×10^{-4}, and it will be smoothly decayed to 0.9 scale every 80k steps. We train 140k steps at pretraining stage and finetuning stage. We adopt data augmentation such as rotation, adding noises and brightness adjustment. The batch size is set to 16.

Experiment. In order to verify the effect of PBRHand on depth estimation, we compare the depth recovery performance on the Stereo dataset [8] using models trained on our PBRHand dataset, RHD and Stereo datasets. Table VI shows the comparison of depth estimation on the Stereo dataset. We can see that the model trained on PBRHand+Stereo achieves the best performance. Fig. 12 shows several qualitative results. We can see that the model pretrained with our PBRHand dataset recovers more accurate depth images.

TABLE VI
PERFORMANCE OF DEPTH ESTIMATION FROM COLOR IMAGE (IN MM) ON STEREO DATASET WITH DIFFERENT TRAINING DATASETS.

Training Data	Stereo	RHD+Stereo	PBRHand+Stereo
MAE ↓	5.877	4.571	**4.541**

D. 3D Hand Pose from Depth Image

In this experiment, we aim to compare the effect of synthetic training datasets with our scanned hand models in PBRHand and a parametric hand model MANO [35] on 3D hand pose estimation. Since the original MANO model does not contain photo-realistic textures to support color based hand pose related tasks, we conduct the comparison using 3D hand pose estimation from depth. During the comparison, we generate two synthetic training datasets PBRHand-Sub and MANO, which both consist of 10 hand models and 1840 hand pose.

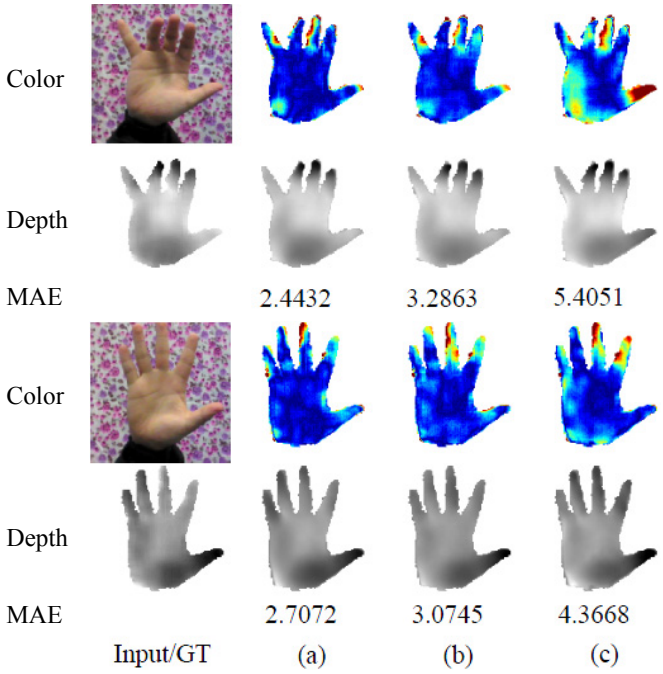


Fig. 12. Depth estimation from color image on Stereo dataset. (a)(b)(c) show the predicted depth and error map of predicted depth using models pretrained on our PBRHand dataset, pretrained on RHD, and without pretraining, respectively. The model pretrained with our PBRHand dataset recovers more accurate depth images.

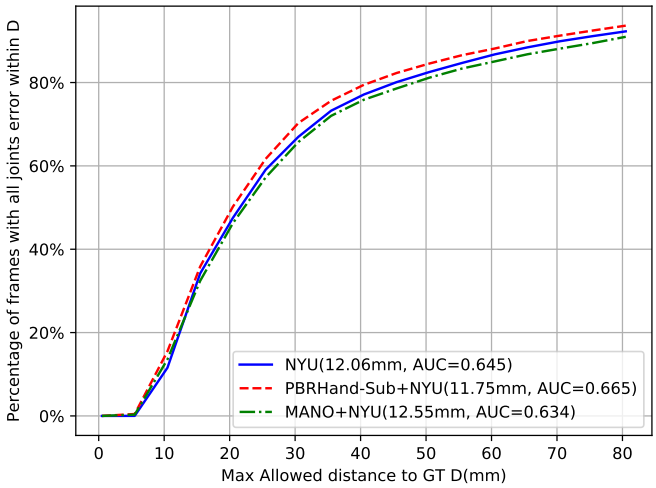


Fig. 13. 3D hand pose results with DenseReg trained on NYU dataset, PBRHand-Sub + NYU dataset, and MANO + NYU dataset.

TABLE VII
PERFORMANCE OF 3D HAND POSE FROM DEPTH IMAGE ON NYU DATASET WITH DIFFERENT TRAINING DATASETS. AUC IS SHOWN IN PERCENTAGE POINTS.

Dataset	NYU	MANO + NYU	PBRHand-Sub + NYU
Mean error (mm) ↓	12.06	12.55	**11.75**
AUC(0-80mm) ↑	64.5	63.4	**66.5**

The PBRHand-Sub dataset, a subset of our full PBRHand dataset, adopts scanned hand shape models, and the hand shape models in the MANO dataset is generated via the MANO hand model [35].

Method. We use the network architecture in DenseReg [36] for 3D hand pose estimation from a single depth. We choose DenseReg [36] as the baseline model, because DenseReg is one of the state-of-the-art networks with the released code. The network adopts both the 2D and 3D properties of a depth to recover 3D hand pose via pixel-wise regression.

Training. We conduct our comparison experiments on NYU dataset [16]. To highlight the difference between scanned hand models and MANO hand models for 3D hand pose estimation, we train DenseReg [36] from scratch on NYU dataset, PBRHand-Sub + NYU dataset, and MANO + NYU dataset. The network are trained with a batch size of 40 and using ADAM solver. The initial learning rate is 0.001, and the exponential decay rate of the momentum is set to $\beta_1 = 0.5$. We donot use data augmentation during the training process.

Experiment. We train with the code released by [36] and test on the NYU hand dataset. We get three DenseReg models trained on NYU dataset, PBRHand-Sub + NYU dataset, and MANO + NYU dataset, and compare their 3D hand pose estimation performance. Fig. 13 and Table VII compare the 3D hand pose results on NYU dataset with different training datasets. The model trained with PBRHand-Sub + NYU dataset achieves better performance than the model trained with MANO + NYU dataset and the model trained on NYU dataset, which indicates that our hand model is more realistic than MANO and the rendered data is more beneficial to the pre-training.

V. MULTI-TASK HAND POSE ESTIMATION FROM COLOR

With the help of our dataset, we further propose a multi-task deep neural network for hand pose estimation, which jointly predicts depth and hand joints in a cascaded fashion. We show how our multi-task model and our synthetic dataset compare with the state-of-the-art models pretrained with other synthetic datasets. Fig. 14 illustrates the basic structure of our network. In the first stage, the network predicts a depth map from input hand image. And in the second stage, the network uses latent features from first stage to predict 3D hand joints. We select depth as an auxiliary task based on the fact that depth is closely related to 3D hand pose estimation [2].

A. Data Preparation

We denote the training samples as $\{(I_i, D_i, J_i)\}_{i=1}^{N}$, where I_i is the i-th detected hand region, D_i and J_i are the corresponding hand depth map and 3D hand joint locations in the camera's coordinate system. Following commonly used data normalization methods [3], the input color image is normalized to $[-1, 1]$. Considering the sizes of human hands, we choose a constant value $\alpha = 100$ mm to normalize depth and 3D hand joints. The depth image is first subtracted by the average depth of the hand region and then normalized with α. For 3D hand joints, we first find the 3D center of the mass (CoM) of the hand region using the depth map. Each hand joint is then normalized by subtracting CoM and divided by α. This CoM is only used to crop the hand but not directly used as input

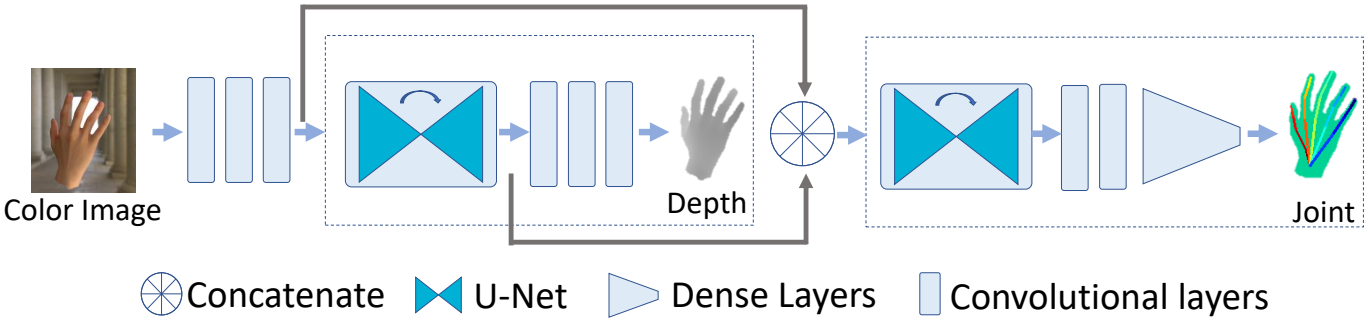


Fig. 14. Our cascaded multi-task network for hand pose estimation from a single color image. The network uses cropped and resized hand regions as input. Our network has two stages, the first stage for depth recovery, and the second stage for hand pose recovery, which iteratively generates depth and hand pose.

of the network in anyway. The network output is a 3D pose relative to a CoM, and the CoM is only required to compare with the ground truth. Hereafter, we denote D_i and J_i to be the normalized depth map and joints, respectively.

B. Cascaded Multi-task Network

Our network takes a color image with the resolution of $128 \times 128 \times 3$ as input. We first use several convolutional blocks to extract *image features*. After that, in the depth prediction stage, we use an U-net [37] to generate *latent depth features*. U-net is well known for its capability to extract both low-level and high-level features and widely used in segmentation, detection and pixel-wise regression. Then the depth features are passed to several convolutional blocks to generate depth map. In the joint prediction stage, the network first concatenates *image features* and *depth features*. And then, another U-net is employed to fuse the input features. Finally, after several convolutional blocks, downsampling and fully connected layers, the network outputs normalized 3D hand joint positions. Our network predicts depth and joints in a cascaded fashion, where joints depends on not only input image features but latent depth features as well. More details about the network can be found in the appendix.

C. Loss Functions

In the training phase, the total loss function for the proposed cascaded network is defined as:

$$L_{total}(\hat{J}, \hat{D}) = L_{pose}(\hat{J}, J^*) + \lambda_d L_{depth}(\hat{D}, D^*) \quad (1)$$

where the loss L_{total} is a combination of hand pose loss L_{pose} and depth recovery loss L_{depth}. The parameter λ_d is the weight of L_{pose} and L_{depth}, and set to 1.0.

a) Loss function for hand pose: We use Euclidean distance to evaluate the difference between the ground truth and estimated 3D hand joints.

$$L_{pose}(\hat{J}, J^*) = \frac{1}{n}\sum_{i=1}^{n} ||\hat{\mathbf{j}}_i - \mathbf{j}_i^*|| \quad (2)$$

where $\hat{J} = \{\hat{\mathbf{j}}_i\}$ and $J^* = \{\mathbf{j}_i^*\}$ are the predicted joint positions and ground truth joints, respectively, n is the total number of joints, which is typically 21 in most datasets.

b) Loss function for depth recovery: Similar to the image regression work [38], we use L_1 norm to evaluate the loss between the ground truth and estimated depth maps on the hand foreground region.

$$L_{depth}(\hat{D}, D^*) = \frac{1}{m}\sum_{i=1}^{m} |M \odot (\hat{\mathbf{d}}_i - \mathbf{d}_i^*)| \quad (3)$$

where $\hat{D} = \{\hat{\mathbf{d}}_i\}$ and $D^* = \{\mathbf{d}_i^*\}$ are the predicted and ground truth depth maps, respectively. M is the hand foreground binary mask, m is the number of valid pixels in the hand foreground binary mask, and $\odot$ represents element-wise multiplication.

D. Implementation Details

Our method focuses on hand pose estimation, where hand regions are detected and cropped following the way in [39], and the cropped color images and depth maps are resized to 128×128 pixels. During the training stage, we perform data augmentation on the training datasets using 2D rotation, scaling for the hand region, depth image and and ground truth hand joints. We use in-plane rotation around z-axis of the camera's coordinate system. The in-plane rotation angle is randomly sampled from the interval [-5, 5] degrees. We keep the aspect and scale hand color images with factors randomly chosen from $[0.84, 1.16]$. Our implementation is based on Tensorflow [40]. We use RMSProp optimizer with 0.001 learning rate, 0.9 weight decay. We use 16 batch size and train on our PBRHand dataset for 10 epochs and finetune on the training set of each evaluating dataset for 20 epochs.

VI. Experiment

In this section, we first compare our multi-task hand pose estimation model with state-of-the-art, then conduct ablation study on our method.

A. Comparison with State-of-the-art methods

We first evaluate our cascaded multi-task network and compare with state-of-the-art methods on the Stereo dataset (Fig. 15 and Table VIII). Our network is pretrained on our PBRHand dataset, and then finetuned on the Stereo training set. Weakly-sup [7], Latent2.5D [32], Spurr [41], Mueller

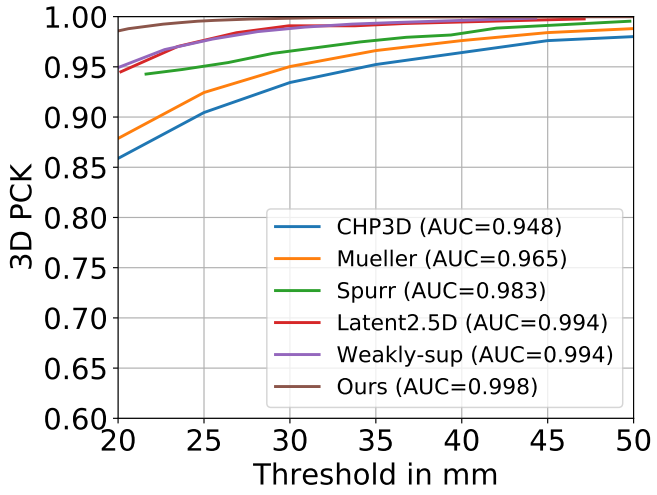


Fig. 15. Comparison of different methods on Stereo dataset. Our method outperforms all other methods and achieves the highest AUC.

et al.[4] and CHP3D [3] are the state-of-the-art color-based hand pose methods using the same experiment setting (i.e. the same training and testing datasets) as our method. For Stereo dataset, the area under the curve (AUC) of our network are higher than the results of the compared methods, Weakly-sup [7], Latent2.5D [32], CHP3D [3], Spurr [41], and Mueller [4]. More evaluations can be found in the supplementary document.

We then use the synthetic RHD dataset to further verify the effectiveness of both the synthetic data and our model. Table VIII shows the comparison results. To compare on the dataset, we pretrain our cascaded model with RHD and our PBRHand, respectively. We use both testing set from RHD and Stereo for evaluation. We first train models on RHD or our dataset, and then finetune for the Stereo dataset. For Stereo dataset, the model pretrained on our PBRHand (6.39mm) achieves better performance than the model pretrained on RHD (7.03mm), which shows our synthetic dataset contributes more to the pretraining. To compare about the models, we pretrain CHP3D [3] and our model using RHD, finetune and test on Stereo datasets. Again, our model (7.03mm) achieves an error 5.17mm lower than [3] (12.2mm), and also performs better than Weakly-sup [7] and Spurr [41]. We also evaluate the performance on RHD testing set with CHP3D + RHD, Ours + RHD, Ours + PBRHand. As shown in Table VIII, the 3D EPE with Ours + RHD (17.35mm) and Ours + PBRHand (17.12mm) is 18.25mm and 18.48mm lower than CHP3D + RHD (35.6mm).

We also evaluate our method on the Dexter+Object dataset and compare to the state-of-the-art Latent 2.5D [32] and Mueller [4]. Table IX shows the numerical evaluations. We can observe that our method trained on RHD+Stereo performs better than Mueller [4] (Ours AUC:0.572, Mueller AUC:0.560). Since Latent 2.5D has not released the code yet, we use the reported scores which is trained on RHD+Stereo. We also train our model on RHD+Stereo, and we observe that our method achieves comparable performance to Latent 2.5D on Dexter+Object dataset (Ours EPE:43.31, AUC:0.572, Latent 2.5D EPE:45.54, AUC:0.570).

TABLE VIII
EPE AND AUC (BETWEEN 20 - 50MM) EVALUATION ON RHD AND STEREO DATASETS. OUR CASCADED NETWORK WITH PRETRAINING ON OUR DATASET SHOWS CONSISTENTLY BETTER PERFORMANCE THAN OTHERS. OURS: OUR CASCADED NETWORK. AUC IS SHOWN IN PERCENTAGE POINTS.

	RHD		Stereo	
	EPE↓	AUC↑	EPE↓	AUC↑
CHP3D+RHD [3]	35.60	67.0	12.2	94.80
Weakly-sup [7]	-	88.7	-	99.4
Spurr [41]	-	84.9	-	98.6
Ours+RHD	17.35	88.93	7.03	99.68
Ours+PBRHand	**17.12**	**90.39**	**6.39**	**99.79**
Ours wo pretrain	17.35	88.93	7.07	99.62

TABLE IX
EPE AND AUC (BETWEEN 0-100MM) ON DEXTER+OBJECT DATASETS. OUR CASCADED NETWORK WITH PRETRAINING ON OUR DATASET SHOWS CONSISTENTLY BETTER PERFORMANCE THAN OTHERS. OURS: OUR CASCADED NETWORK. '*' MEANS THE EPE AND AUC FOR THE ABSOLUTE 3D POSES AS LATENT 2.5D [32]. AUC IS SHOWN IN PERCENTAGE POINTS.

	EPE↓	AUC↑
Mueller [4]	-	56.0
Latent 2.5D [32]	45.54*	57.00*
Ours	43.31	57.20

Finally, we also evaluate our method on Hands-2019 online challenge [31], which is especially challenging due to occlusions of object. For Hands-2019, we submit our hand pose results online, and adopt the evaluation protocol in [31] to calculate mean joint errors for four splits of the test set of Hands-2019 (whether the test split has hand shapes or objects present in the training set), i.e. EXTRAP., INTERP., OBJECT, and SHAPE (refer to [31] for details). EXTRAP., OBJECT and SHAPE are three data splits to verify the generalization power of the model for hand shapes and objects, and EXTRAP. is the key test split for evaluation in Hands-2019 competition. Table X shows the comparison to the state-of-the-art methods [42], [43] (submitted by 'User:yhasson' at 2019-10-09 and 'User:lin84' at 2019-10-07). Their methods train networks on a large-scale synthetic dataset of hands grasping objects [42] or hands with neighboring objects [43]. Compared to the synthetic data used in [42], [43], our PBRHand dataset does not contain objects, thus it has domain gap to Hands-2019 dataset. Our method pretrained on PBRHand ('User:Fractality') can still achieve better performance on the data splits of EXTRAP., INTERP. and OBJECT than [42]. Compared to [43], our method performs better in all the four test data splits. We also observe that our cascaded network performs better when pretrained on our PBRHand compared to RHD dataset. These results indicate the generalization ability our model for hand shape and object.

B. Ablation Study

To further understand the task and the network, we conduct several comparison experiments using our multi-task network and our single task network with/without pretrain.

a) Does pre-training help?: From Table XI, we can observe that by pretrained on our PBRHand dataset the

TABLE X
COMPARISON OF MEAN JOINT ERRORS (IN MM) ON HANDS-2019 ONLINE CHALLENGE [31]. OUR CASCADED NETWORK PRETRAINED ON PBRHAND PERFORMS BETTER THAN OTHERS. OURS: OUR CASCADED NETWORK.

	EXTRAP.↓	INTERP.↓	OBJECT↓	SHAPE↓
User:yhasson [42]	38.42	7.38	31.82	**15.61**
User:lin84[43]	31.51	19.15	30.59	23.47
Ours	36.70	14.23	40.51	32.88
Ours+PBRHand (User:Fractality)	**28.81**	**6.61**	**22.88**	20.76
Ours+RHD	30.31	11.24	30.59	24.63

TABLE XI
COMPARISON OF 3D EPE (IN MM) AND AUC (BETWEEN 20MM TO 50MM) ON STEREO AND RHD DATASETS USING OUR SINGLE TASK AND CASCADED MULTI-TASK METHODS WITH OR WITHOUT PRETRAIN. AUC IS SHOWN IN PERCENTAGE POINTS.

	Stereo hand dataset		RHD dataset	
	EPE↓	AUC↑	EPE↓	AUC↑
single wo pretrain	8.93	98.70	18.77	86.81
single w pretrain	**7.13**	**99.64**	**18.21**	**87.90**
cascade wo pretrain	7.07	99.62	17.35	88.93
cascade w pretrain	**6.39**	**99.79**	**17.12**	**90.39**

average errors of hand joints with cascaded network (with finetune) drop by 0.68mm and 0.23mm for Stereo and RHD datasets. From Table XII, considering the same network with different training data, we can observe that the networks trained on PBRHand provide consistently better performance on Dexter+Object dataset. We can see that pretraining on our PBRHand dataset is an effective way to enhance the performance of color based hand pose estimation for single task or multi-task networks.

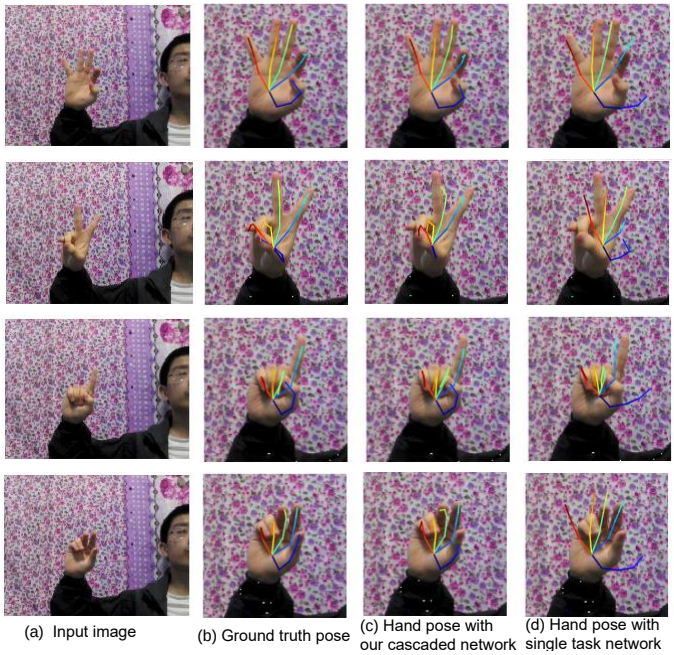


Fig. 16. Examples of results with our method on Stereo dataset. (a) input image. (b) the ground truth pose. (c) hand pose with our cascaded network. (d) hand pose with single joint network wo depth cascade.

b) Does multi-task help?: In order to evaluate the multi-task networks for hand pose estimation, we compare the baseline single-task network, which shares the same network architecture as the cascaded task network. The difference is that the single task network does not adopt auxiliary depth recovery task and does not use the depth loss. Fig. 16 and Table XI shows the accuracy of 3D hand joints with and without multi-task supervisions. We can observe that results with our cascaded network is consistently better than results with single task network. For Stereo and RHD, the average errors of hand pose using cascaded network with pretraining drop about 2.72mm, 1.09mm compared to the single task network with pretraining, and the average errors of the cascaded network without pretraining drop about 1.86mm, 1.42mm, compared to the single task network without pretraining. For Dexter+Object dataset (See Table XII), by comparing networks trained on the same datasets, we can observe that our cascaded multi-task network performs better than others. First, it is generally better than the single joint task network (except trained only on Stereo dataset, which might be caused by the limited size and diversity of the dataset). One possible explanation is that multi-task cascaded training can provide more cues (*e.g.*depth features) and improve the feature extraction on early layers (color image features in our network). We see that our cascaded multi-task network is useful for the training with and without pretraining.

TABLE XII
COMPARISON OF MEAN 3D EPE (IN MM) AND AUC (BETWEEN 0MM TO 100MM) ON DEXTER+OBJECT DATASET [10]. WE USE DIFFERENT TRAINING DATA COMBINATION AS WELL AS OUR SINGLE TASK AND CASCADED MULTITASK NETWORKS. FROM THE TABLE WE CAN OBSERVE THAT OUR CASCADED NETWORK TRAINED ON PBRHAND ONLY PERFORMS THE BEST (HIGHEST AUC AND LOWEST EPE). AUC IS SHOWN IN PERCENTAGE POINTS.

	single		cascade	
	EPE↓	AUC↑	EPE↓	AUC↑
Stereo	50.04	51.40	52.35	48.50
RHD	49.05	51.20	48.80	51.40
RHD + Stereo	51.78	50.40	43.31	57.20
PBRhand	43.30	57.40	**40.70**	**59.53**

c) Feature cascade vs. input/output cascade: The proposed cascaded network uses *image features* and *depth features* instead of input image and output depth. We evaluate the effects of different cascade inputs on hand pose estimation performance. We compare the following settings: $RGB + D$, $RGB_{feature} + D$, $RGB + D_{feature}$, D, $D_{feature}$ and the proposed $RGB_{feature} + D_{feature}$. RGB and $RGB_{feature}$ mean the input color image and the feature extracted with first three convolution layers of our network. D and $D_{feature}$ are the recovered depth with depth recovery network and the depth-aware feature extracted with the first U-net in Fig. 14. Table XIII shows that feature cascade would produce the best result for joint prediction.

More ablation studies can be found in the supplementary document.

C. Runtime and Qualitative Experiments

Our network contains about 26.7M parameters, and the testing time is 180 fps on a computer with a NVIDIA Titan X GPU, which also has the potential to run on embedding systems in real-time.

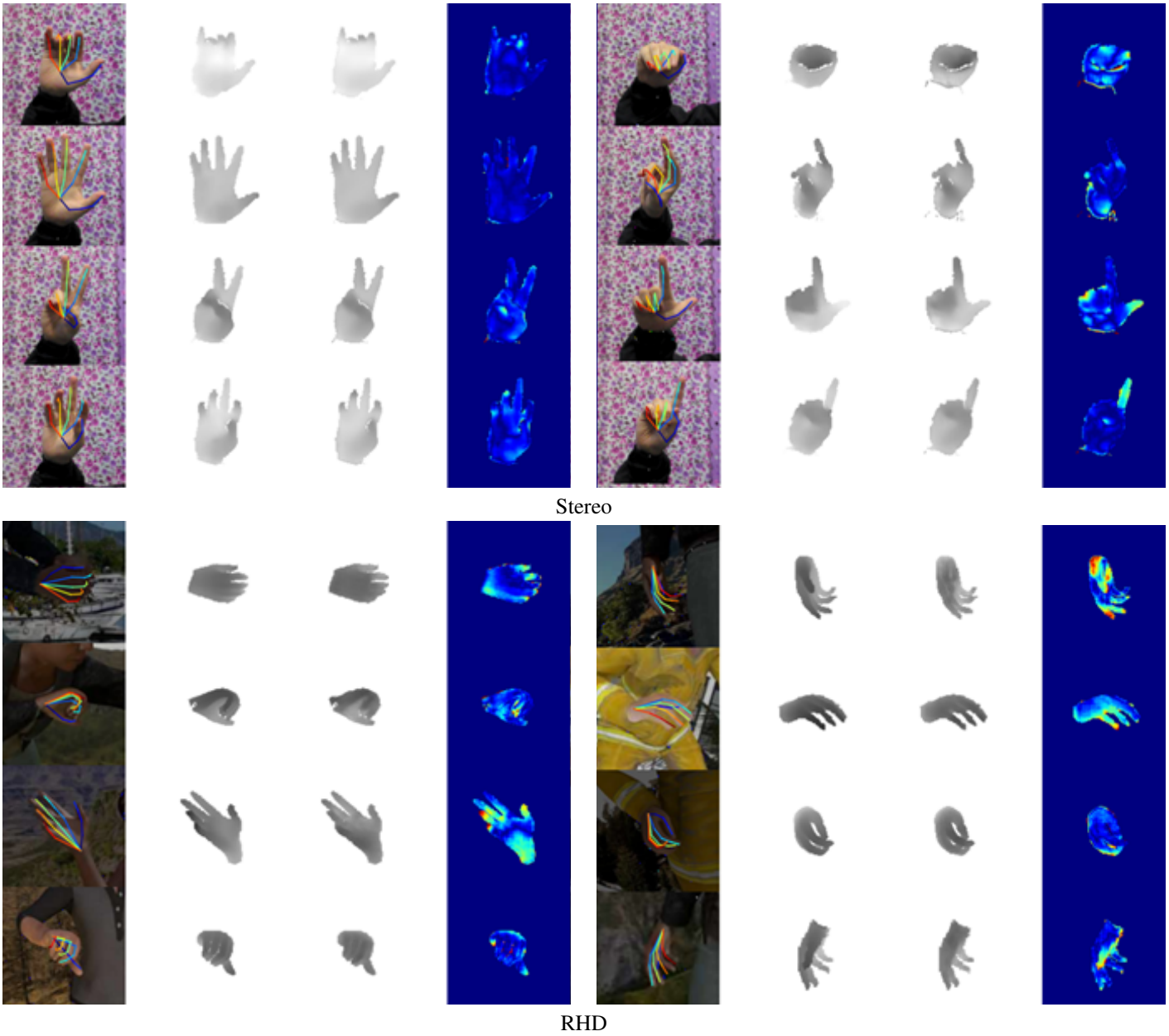


Fig. 17. Qualitative results on Stereo and RHD datasets. From left to right: input color image and the predicted joint; ground truth depth; predicted depth image; difference between ground truth and the prediction.

TABLE XIII
CASCADE EVALUATION. WE TRIED DIFFERENT COMBINATION OF DATA THAT CASCADED FOR JOINT ESTIMATION. $RGB_{feature} + D_{feature}$ PRODUCES THE BEST RESULT, WHICH IS THE CHOOSEN NETWORK STRUCTURE. ONLY PASS D TO JOINT NETWORK WOULD IMPROVE DEPTH PREDICTION, BUT CANNOT PRODUCE THE BEST JOINT PREDICTION. AUC IS SHOWN IN PERCENTAGE POINTS.

	Depth		Joint	
Cascade type	RMSE ↓	MAE ↓	EPE ↓	AUC ↑
$RGB + D$	9.08	7.59	8.10	99.17
$RGB_{feature} + D$	9.52	7.41	8.13	99.32
$RGB + D_{feature}$	15.46	12.82	8.55	98.96
D	**8.42**	**6.63**	7.78	99.47
$D_{feature}$	8.93	6.94	7.13	99.59
$RGB_{feature} + D_{feature}$	8.68	6.63	**7.07**	**99.68**

Fig. 17 shows our results on Stereo and RHD dataset, with our model pretrained on PBRHand and finetuned on Stereo and RHD training dataset. More qualitative results using captured real data can be found in the supplementary document.

VII. CONCLUSION

In this paper, we construct a large-scale hand pose dataset with photo-realistic rendering color images and various ground truths. Based on the dataset, we first verify on three hand pose related tasks, and find that pretraining on our photo-realistic rendering dataset can enhance the performance of the state-of-the-art models. With the help of our dataset, we further design a cascaded multi-task network to recover 3D hand pose from a color image with an intermediate depth supervision. We find that our dataset benefits the model pretraining by providing

various ground truths and our cascaded network can improve the performance, both enhancing the pose estimation results.

APPENDIX
DETAILS OF OUR NETWORK ARCHITECTURE

Fig. 18 illustrates the details for our network architecture.

REFERENCES

[1] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[2] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2.2m benchmark: Hand pose dataset and state of the art analysis," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proceedings of International Conference on Computer Vision*, 2017.

[4] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric rgb-d sensor," in *Proceedings of International Conference on Computer Vision*, 2017.

[5] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser, "Physically-based rendering for indoor scene understanding using convolutional neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] J. Shi, Y. Dong, H. Su, and X. Y. Stella, "Learning non-lambertian object intrinsics across shapenet categories," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," *Proceedings of European Conference on Computer Vision*, 2018.

[8] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang, "3d hand pose tracking and estimation using stereo matching," *arXiv preprint arXiv:1610.07214*, 2016.

[9] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: Dataset for markerless capture of hand pose and shape from single rgb images," in *IEEE International Conference on Computer Vision*, 2019.

[10] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *Proceedings of European Conference on Computer Vision*, 2016.

[11] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proceedings of ACM Conference on Human Factors in Computing Systems*, 2015.

[12] A. Wetzler, R. Slossberg, and R. Kimmel, "Rule of thumb: Deep derotation for improved fingertip detection," *arXiv preprint arXiv:1507.05726*, 2015.

[13] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla, "Large-scale multiview 3d hand pose dataset," *arXiv preprint arXiv:1707.03742*, 2017.

[14] Blender, "Blender," 2018, http://www.blender.org.

[15] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.

[16] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics*, vol. 33, no. 5, 2014.

[17] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[18] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3d training data for fine hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4957–4965.

[19] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[20] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang, "A hand pose tracking benchmark from stereo matching," in *Proceedings of IEEE International Conference on Image Processing*, 2017.

[21] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[22] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[23] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, vol. 126, no. 9, pp. 942–960, 2018.

[24] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[25] G. Rogez and C. Schmid, "Mocap-guided data augmentation for 3d pose estimation in the wild," in *Advances in Neural Information Processing Systems*, 2016.

[26] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," *arXiv preprint arXiv:1712.01057*, 2017.

[27] Artec, "Artec eva," https://www.artec3d.com.

[28] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[29] Wikipedia, "Uv mapping," 2018, https://en.wikipedia.org/wiki/UV_mapping.

[30] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Transactions on graphics*, vol. 26, no. 3, p. 72, 2007.

[31] "Task 3 of hands19 challenge: , rgb-based 3d hand pose estimation while interacting with objects," 2019. [Online]. Available: https://competitions.codalab.org/competitions/21116

[32] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz, "Hand pose estimation via latent 2.5 d heatmap regression," *Proceedings of European Conference on Computer Vision*, 2018.

[33] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[34] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[35] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, 2017.

[36] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3d regression for hand pose estimation," *arXiv preprint arXiv:1711.08996*, 2017.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015.

[38] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[39] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3d hand pose estimation," in *ICCV workshop*, 2017.

[40] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[41] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[42] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[43] L. Yang, S. Li, D. Lee, and A. Yao, "Aligning latent spaces for 3d hand pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[44] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of ICML*, 2013.

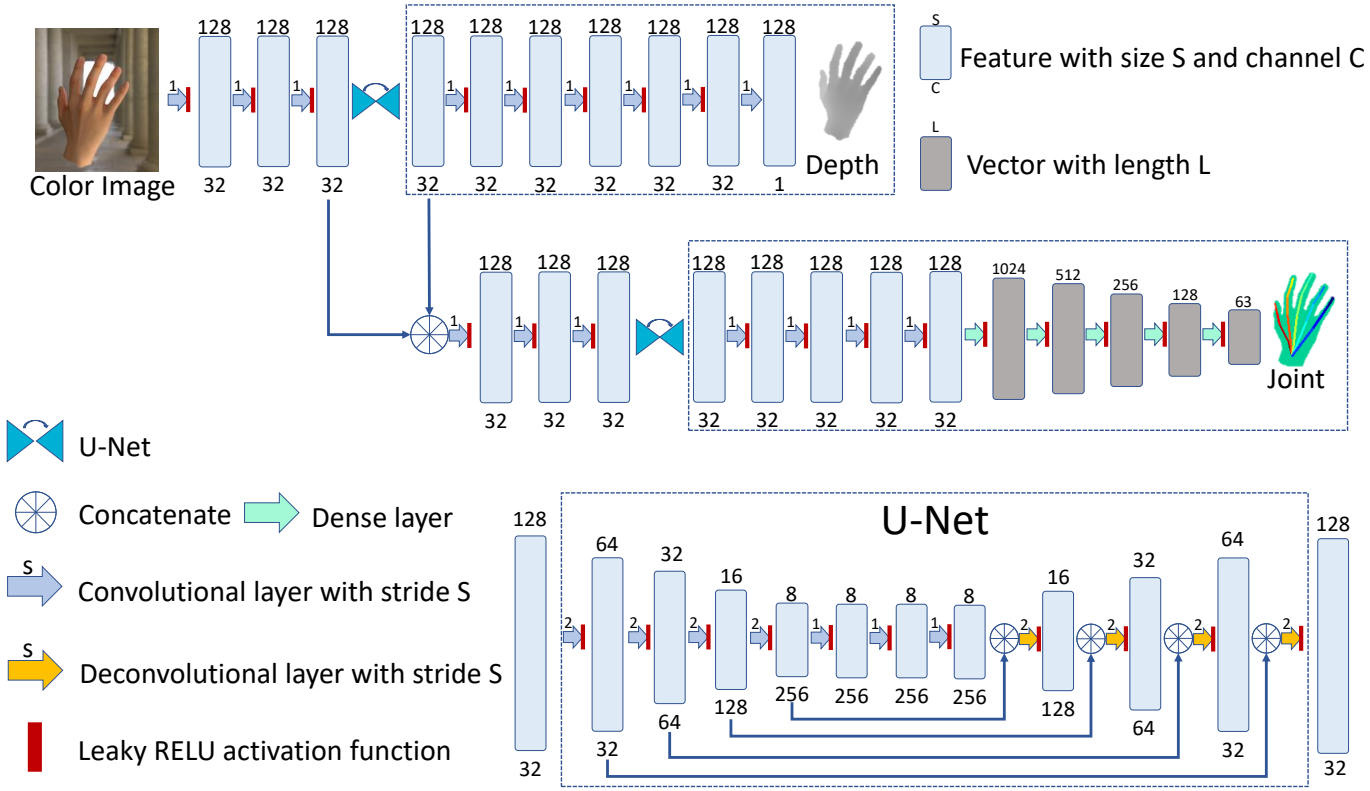


Fig. 18. Full details of our multi-task cascaded network. Kernel size of the convolutional layers and deconvolutional layers is 3. Alpha value of the Leaky RELU [44] activation function is 0.2. To avoid zero patterns caused by up-sampling, each deconvolutional layer is followed by two convolutional layers.