HOnnotate: A method for 3D Annotation of Hand and Objects Poses Supplementary Material

1. Hand Pose Estimation from Color Single Image

Fig. 1 shows the architecture of our hand pose estimator from a single frame. Given an image of the hand centered in the image window, we first extract features using the convolutional layers of VGG [5], and then similar to [2] using a multi-stage CNN, we predict heatmaps for the 2D hand joint locations and finally joint direction vectors with respect to wrist joint. The hand detection can be done using segmentation as described in Section 6.

2. Hand Pose Estimation for Hand Interaction with Unseen Objects

Knowing the objects in advance can help to improve the performances of the estimated 3D hand pose while hand interacts with objects, however, in practice, the hand can manipulate any arbitrary objects. We have tested our hand pose estimator trained on our annotations, and tested on sequences where a hand is manipulating objects not present in the annotated images. As shown in Fig. 6, our pose estimator performs well on these sequences.

3. Hand Shape Estimation

The MANO hand shape parameters $\beta \in \mathbb{R}^{10}$ were estimated for each human manipulator in our HO-3D dataset. The shape parameters are estimated from a sequence Φ of hand only poses using a method similar to [6] in two steps. More exactly, the pose of hand p_h^t in the sequence is first estimated for each frame t using a mean pose β_{mean} as $\hat{p}_h^t = \arg\min_{p_h} E_H(p_h, \beta_{mean})$, where,

$$E_H(p_h, \beta_{mean}) = E_{\mathcal{D}}(p_h, \beta_{mean}) + \epsilon E_{\text{joint}}(p_h) + \eta E_{\text{tc}}(p_h, p_h^{t-1}, p_h^{t-2}). \quad (1)$$

$E_{\mathcal{D}}(p_h, \beta_{mean})$ represents the data term defined in Eq. 2 of the paper where hand is rendered with pose parameters p_h and shape parameters β_{mean}. E_{joint} and E_{tc} are explained in Section 3.2 of the paper. At each frame, the pose parameters are initialized with p_h^{t-1}. The personalized hand shape parameters are then obtained as,

$$\beta^* = \arg\min_{\beta} \sum_{t \in \Phi} \min_{p_h^t} E_H(p_h^t, \beta), \quad (2)$$

where the pose parameters are initialized with the values obtain in the first step ($\hat{p}_h^t$).

Joint	Index	Middle	Pinky	Ring	Thumb
MCP	(0.00, 0.45) (-0.15, 0.20) (0.10, 1.80)	(0.00, 0.00) (-0.15, 0.15) (0.10, 2.00)	(-1.50, -0.20) (-0.15, 0.60) (-0.10, 1.60)	(-0.50, -0.40) (-0.25, 0.10) (0.10, 1.80)	(0.00, 2.00) (-0.83, 0.66) (0.00, 0.50)
PIP	(-0.30, 0.20) (0.00, 0.00) (0.00, 0.20)	(-0.50, -0.20) (0.00, 0.00) (0.00, 2.00)	(0.00, 0.00) (-0.50, 0.60) (0.00, 2.00)	(-0.40, -0.20) (0.00, 0.00) (0.00, 2.00)	(-0.15, 1.60) (0.00, 0.00) (0.00, 0.50)
DIP	(0.00, 0.00) (0.00, 0.00) (0.00, 1.25)	(0.00, 0.00) (0.00, 0.00) (0.00, 1.25)	(0.00, 0.00) (0.00, 0.00) (0.00, 1.25)	(0.00, 0.00) (0.00, 0.00) (0.00, 1.25)	(0.00, 0.00) (-0.50, 0.00) (-1.57, 1.08)

Table 1: Empirically derived minimum and maximum values for the joint angle parameters used in our implementation.

4. Joint Angle Constraints

The maximum and minimum limits on the joint angle parameters used in Eq. (8) of the paper are provided in Table 1.

5. Point Cloud from Multiple Cameras

The E_{3D} term in Section 3.2 of the paper uses the combined point cloud P from all the RGB-D cameras. Let P_c denote the point cloud corresponding to camera c and M_{c_1,c_2} denote the relative pose between two cameras c_1 and c_2. The consolidated point cloud P is then obtained as,

$$P = [P_0,\ M_{c_1,c_0} \cdot P_1,\ M_{c_2,c_0} \cdot P_2, ...,\ M_{c_N,c_0} \cdot P_N], \quad (3)$$

where $[\cdot,\cdot]$ represents concatenation of point clouds.

6. Hand-Object Segmentation Network

The segmentation maps for the hand and object are obtained from a DeepLabV3 [3] network trained on synthetic images of hand and objects. The synthetic images are obtained by over-laying and under-laying images of hands on

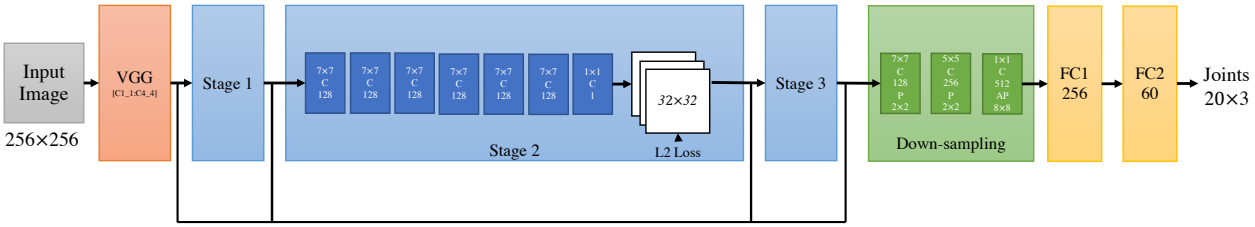

Figure 1: Architecture of our hand pose estimator from a single color image. Given an input image of hand centered in the image, we extract the features using the convolutional layers of VGG [5] (`Conv1_1` to `Conv4_4`). Similarly to [2], we then predict heatmaps for the joint locations in multi-stages. The architecture for the different stages are all the same. `C` denotes a convolutional layer with the number of filters and the filter size inscribed; `FC`, a fully-connected layer with the number of neurons; `P` and `AP` denote max-pooling and average pooling with their sizes, respectively.

Figure 2: Synthetic training images used for training the hand-object segmentation network.

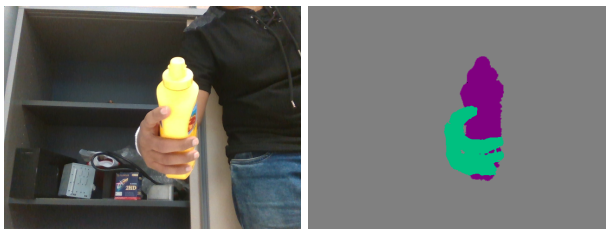

Figure 3: Example of hand and object segmentation obtained with DeepLabV3. Left: input image; Right: hand (green) and object (purple) segmentation.

images of objects at random locations and scales. We use the object masks provided by [1]. The segmented hands were obtained using an RGB-D camera by applying simple depth thresholding. We also use additional synthetic hand images from the RHD dataset [9]. A few example images from the training data are shown in Fig. 2. We use 100K training images with augmentations. Fig. 3 shows segmentation of hand and object using the trained DeepLabV3 network.

7. Automatic Initialization

As explained in Section 4.1 of the paper, a keypoint prediction network based on convolutional pose machine [7] is used to obtain initialization for hand poses. Such a network is trained with our initial hand+object dataset of 15,000 images together with images from hand-only PAN [4] dataset.

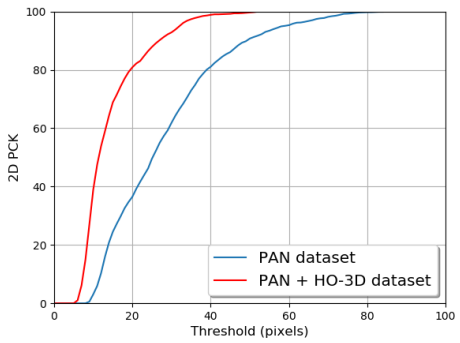

Figure 4: Accuracy of keypoint prediction, described in Section 4.1 of the paper when trained with PAN [4] dataset alone and PAN + our annotations. The accuracy is measured in percentage of correct 2D keypoints given a threshold. Only 15,000 images from our HO-3D dataset are used in training. Due to the presence of object occlusions, a network trained on hands-only dataset is less accurate in predicting keypoints when compared with a network trained with hand+object data.

Figure 4 compares the accuracy of network in predicting keypoints in hand-object interaction scenarios when trained with hands-only dataset and hands+object dataset. Our initial HO-3D dataset helps in obtaining a more accurate network for predicting keypoints and hence results in better initialization.

8. Dataset Details

We annotated 80,000 frames of 65 sequences hand-object interaction of 10 persons with different hand shape. On average there are 1200 frames per sequences. 18 sequences are captured and annotated in a single camera, and 47 sequences for the multi-Camera setup.

Figure 5: 10 objects of the YCB dataset [8] that we use for our dataset HO-3D.

Hand+Object. The participants are asked to perform actions with objects. The grasp poses vary between frames in a sequence in the multi-camera setup and remain almost rigid in the single camera setup.

Participants. The participants are between 20 and 40 years old, 7 of them are males and 3 are females. In total, 10 hand shapes are considered for the annotations.

Objects. We aimed to choose 10 different objects from the YCB dataset [8] that are used in daily life. As shown in Fig. 5, we have a wide variety of sizes such as large objects (e.g. Bleach) that cause large hand occlusion, or the objects that make grasping and manipulation difficult (e.g. Scissors), while these are not the case in the existing hand+object datasets.

Multi-Camera Setup. We use 5 calibrated RGB-D cameras, in our multi-camera setup. The cameras are located at different angles and locations. Our cameras are synchronized with a precision of about 5 ms. The scenes are cluttered with objects, and the backgrounds vary between scenes.

Figs. 7 and 8 show some examples of the 3D annotated frames for both hand and object from our proposed dataset, HO-3D.

References

[1] YCB Benchmarks Object and Model Set. http://ycbbenchmarks.org/. 2

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *CVPR*, 2017. 1, 2

[3] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*, abs/1706.05587, 2017. 1

[4] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, pages 8320–8329, 2018. 2

[5] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014. 1, 2

[6] D. J. Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. Fits Like a Glove: Rapid and Reliable Hand Shape Personalization. In *CVPR*, 2016. 1

[7] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *CVPR*, 2016. 2

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *RSS*, 2018. 3

[9] C. Zimmermann and T. Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In *ICCV*, 2017. 2

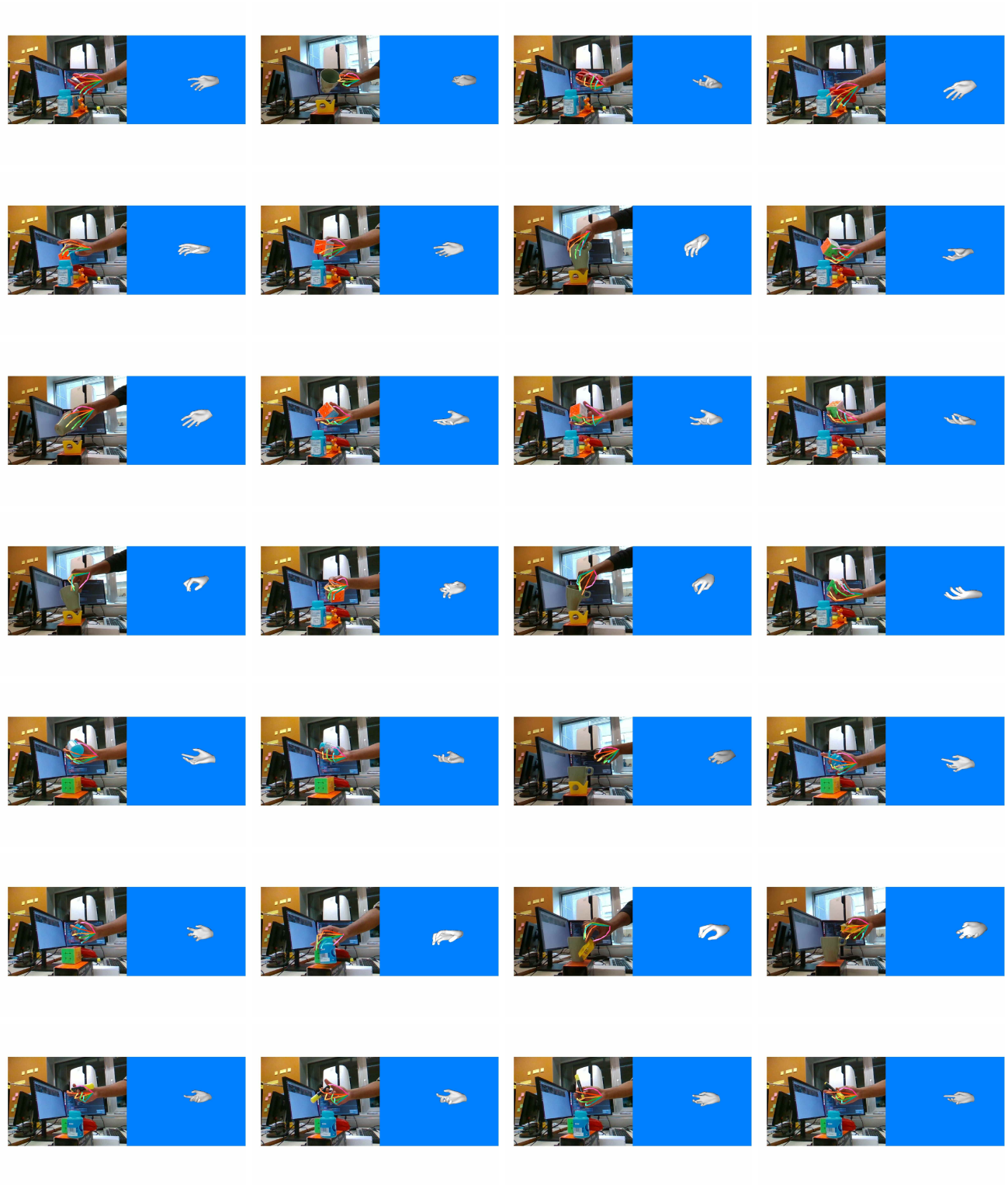

Figure 6: Qualitative results of 3D hand pose estimation of hand manipulating unseen objects. Our pose estimator trained on the HO-3D dataset is still able to predict accurate 3D poses when interacting with new objects.

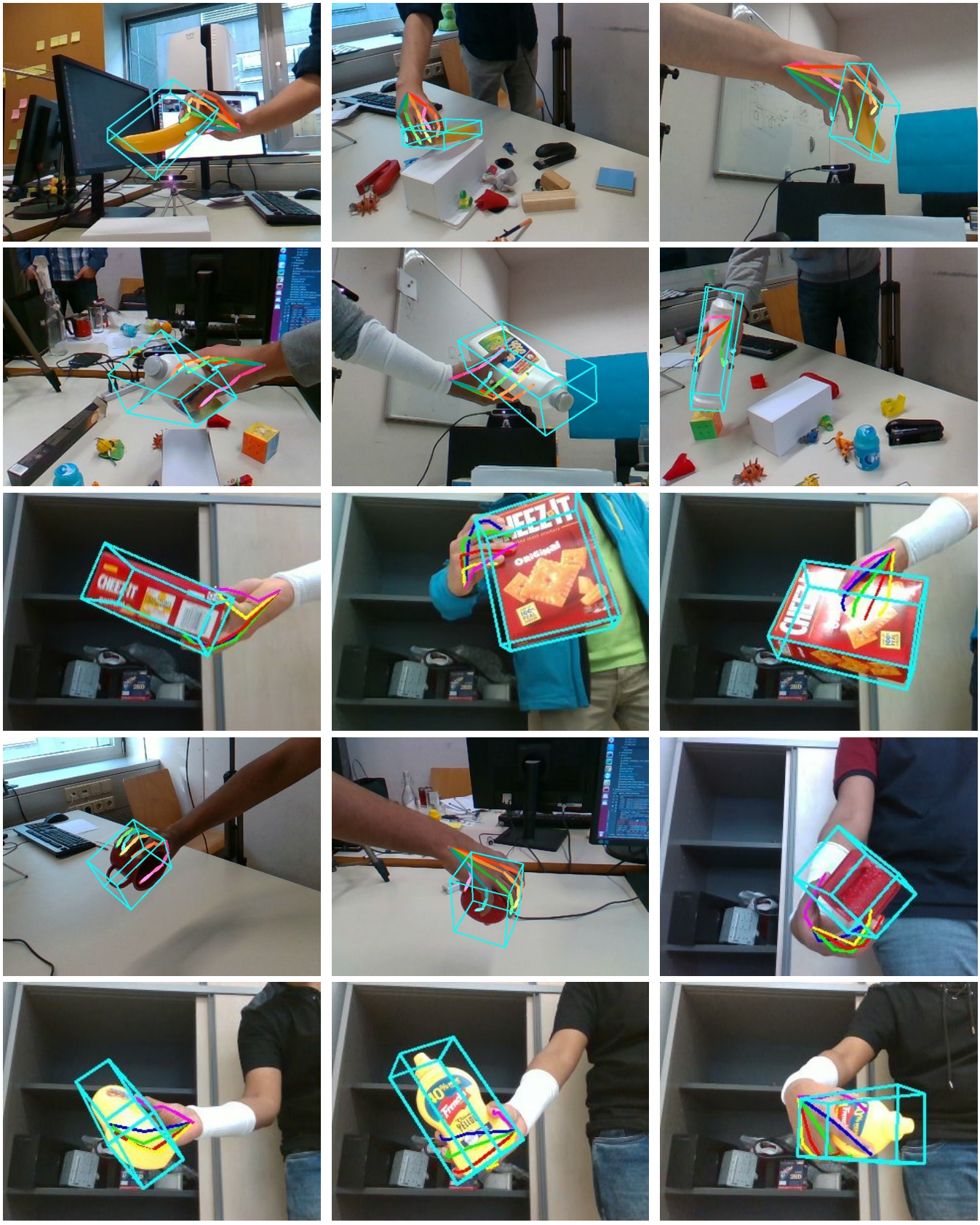

Figure 7: Some examples of the 3D annotated frames for both hand and object from our proposed dataset, HO-3D.

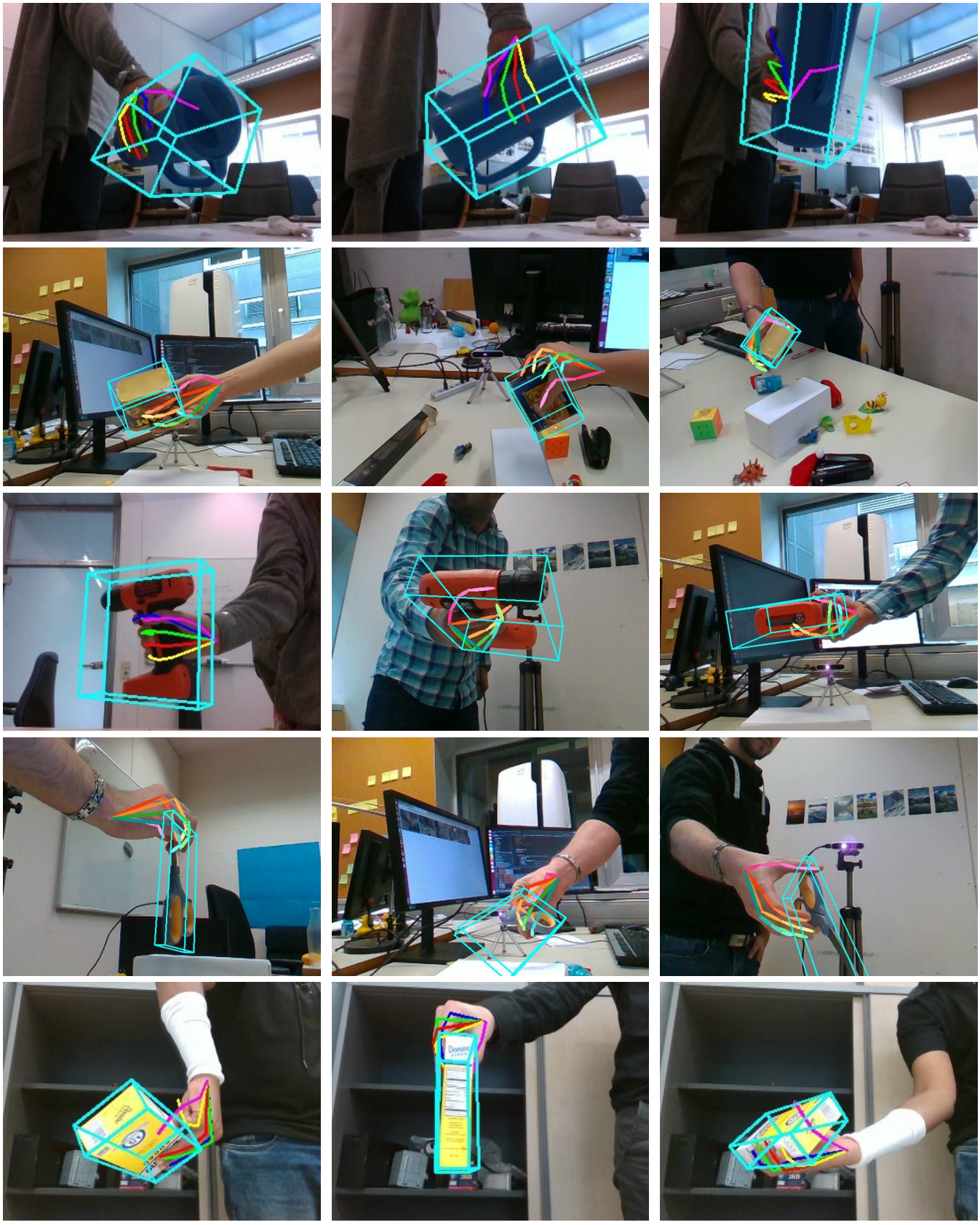

Figure 8: Some examples of the 3D annotated frames for both hand and object from our proposed dataset, HO-3D.