

Supplementary Materials: Monocular Human-Object Reconstruction in the Wild

Anonymous Authors

1 FORMULA CORRECTION

Equation (9) is corrected as

$$\log p(\mathbf{X}_{2.5D}|\mathbf{I}) = \log q(\mathbf{z}) - \log \left| \det \frac{\partial \mathcal{F}}{\partial \mathbf{X}_{2.5D}} \right|,$$

and equation (13) is corrected as

$$\mathcal{L}_{\text{prior}} = - \sum_{i=1}^m \log p(\hat{\mathbf{X}}_{2.5D}^{(i)}|\mathbf{I}).$$

2 METHOD DETAILS

2.1 Top- k Nearest Neighbor Grouping

Given the 2D image dataset $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_m\}$, extract the 2D human-object keypoints $\Pi_\rho(\mathbf{X}_{3D})$ and camera pose $\rho = (\mathbf{R}_{\text{cam}}, \mathbf{t}_{\text{cam}})$ for each image and obtain the intermediate representation $\mathbf{X}_{2.5D}$ according to the left side of equation (6) and equation (7) to form the 2D human-object keypoint dataset $\{\mathbf{X}_{2.5D}^{(1)}, \mathbf{X}_{2.5D}^{(2)}, \dots, \mathbf{X}_{2.5D}^{(m)}\}$. The top- k nearest neighbor grouping algorithm takes this 2D keypoint dataset as input and outputs the neighbor index set for each image $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$. The detailed algorithm is shown in algorithm 1. We state by initializing the neighbors randomly (line 1), then iteratively update the neighbor set by comparing the images that have common neighbors (lines 6-20). In each iteration, the neighbor set for p -th image is updated if the q -th image has a smaller distance to the cluster of p -th image. We also compare the similarity between the q -th image and the items in the cluster of p -th image to ensure the diversity of viewpoints in the cluster of p -th image. The distance $d(\mathbf{X}_{2.5D}^{(i)}, \mathbf{X}_{2.5D}^{(j)})$ is calculated according to the equation (11). After getting the neighbor index set $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$, we compute the cluster for p -th image as

$$\mathcal{G}_p = \left\{ \left(\Pi_{\rho_i}(\mathbf{X}_{3D}), \rho_i, d(\mathbf{X}_{2.5D}^{(p)}, \mathbf{X}_{2.5D}^{(i)}) \right) \mid i \in \mathcal{N}_p \right\}.$$

These clusters are used to train the normalizing flow by minimizing the equation (10).

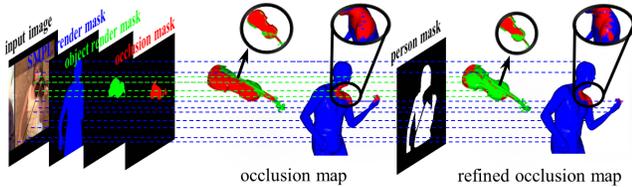


Figure 1: The process of acquiring the occlusion maps.

2.2 Mean Occlusion Map

To get the occlusion map for each image, as shown in figure 1, we first render the SMPL mesh and the object mesh onto the image plane to get the occlusion mask of the image. The points that fall

Algorithm 1 Top- k nearest neighbor grouping with 2D human-object keypoints.

Input: The 2D keypoint dataset $\{\mathbf{X}_{2.5D}^{(1)}, \mathbf{X}_{2.5D}^{(2)}, \dots, \mathbf{X}_{2.5D}^{(m)}\}$, number of iterations n , number of neighbors k , similarity threshold s .
Output: The neighbor index set for each image $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$.

```

1: Randomly initialize the neighbor index set for each image
    $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$ .
2: for  $p \leftarrow 1$  to  $m$  do
3:    $\mathcal{N}'_p \leftarrow \{i \mid p \in \mathcal{N}_i\}$ 
4: end for
5:  $\text{iter} \leftarrow 1$ 
6: while  $\text{iter} \leq n$  do
7:   for  $t \leftarrow 1$  to  $m$  do
8:     for  $p \in \mathcal{N}_t \cup \mathcal{N}'_t, q \in \mathcal{N}_t \cup \mathcal{N}'_t$  do
9:        $d_{\max} \leftarrow \max_{i \in \mathcal{N}_p} \frac{1}{k} \sum_{j \in \mathcal{N}_p} d(\mathbf{X}_{2.5D}^{(i)}, \mathbf{X}_{2.5D}^{(j)})$ 
10:       $i_{\max} \leftarrow \arg \max_{i \in \mathcal{N}_p} \frac{1}{k} \sum_{j \in \mathcal{N}_p} d(\mathbf{X}_{2.5D}^{(i)}, \mathbf{X}_{2.5D}^{(j)})$ 
11:       $d_q \leftarrow \frac{1}{k} \sum_{i \in \mathcal{N}_p} d(\mathbf{X}_{2.5D}^{(i)}, \mathbf{X}_{2.5D}^{(q)})$ 
12:       $s_{\max} \leftarrow \max_{i \in \mathcal{N}_p} \|\mathbf{X}_{2.5D}^{(i)} - \mathbf{X}_{2.5D}^{(q)}\|$ 
13:      if  $d_q < d_{\max}$  and  $s_{\max} < s$  then
14:        Delete  $i_{\max}$  from  $\mathcal{N}_p$  and delete  $p$  from  $\mathcal{N}'_{i_{\max}}$ .
15:         $\mathcal{N}_p \leftarrow \mathcal{N}_p \cup \{q\}, \mathcal{N}'_q \leftarrow \mathcal{N}'_q \cup \{p\}$ .
16:      end if
17:    end for
18:  end for
19:   $\text{iter} \leftarrow \text{iter} + 1$ 
20: end while
21: return  $\{\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m\}$ 

```

into the occlusion region are treated as under occlusion. We further decide whether the front surface is under occlusion or the back surface is under occlusion according to the mask of the person. Here we refer to the front surface as the surface close to the image plane, while the back surface as the surface away from the image plane. The vertex on the front surface of SMPL mesh is treated as under occlusion if its corresponding projected 2D points satisfy the following conditions: (1) it falls into the occlusion region on the image plane, and (2) the person mask at its position is zero. The vertex on the back surface is treated as under occlusion if its corresponding projected 2D points satisfy the following conditions: (1) it falls into the occlusion region on the image plane, and (2) the person mask at its position is one. The conditions for the object are defined similarly. This occlusion information is very valuable to help us to narrow down the contact region. After getting the occlusion map for each image, we average them to get the mean occlusion map for each object. In figure 2, we show the averaged occlusion maps for each object. We can see that the mean occlusion map closely resembles the contact map. This suggests that the

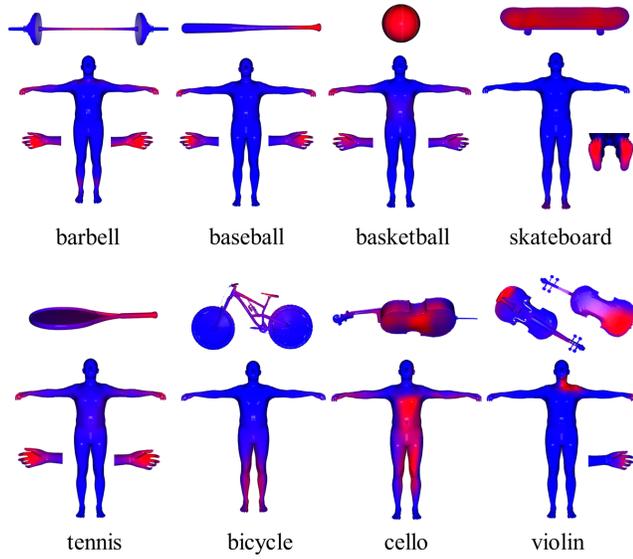


Figure 2: Mean occlusion maps for each object. The occlusion regions are highlighted using red color.

occlusion maps can serve as an effective substitute for contact maps in situations where direct contact information is not available.

3 DATASET DETAILS

The annotation pipeline for the WildHOI dataset is depicted in figure 6. We use the annotations generated by this pipeline to obtain camera 6D pose $\rho = \{\mathbf{R}_{\text{cam}}, \mathbf{t}_{\text{cam}}\}$, 2D human-object keypoints $\Pi_{\rho}(\mathbf{X}_{3\text{D}})$, and the occlusion map $\{c_h, c_o\}$ for each image, which is utilized to train the normalizing flow. To validate our method, we manually annotate a small fraction of images to form the test dataset. The statistics of the WildHOI dataset are shown in table 1. In this reconstruction process, two crucial details are missing: object 6D pose annotation and human-object spatial relationship annotation. Both of these aspects will be elaborated following.

3.1 Object 6D Pose Annotation

Object Keypoint Set Directly label the location and orientation of the object in the image is time-consuming and laborious. One common way to annotate the 6D pose of the object is annotating the 2D position of the predefined 3D keypoints in the image plane to build up the 2D-3D corresponding and using the RANSAC/PnP algorithm to solve the 6D pose of the object. However, due to the various textures of objects in the wild and the occlusion between the human and the object, it is very challenging to predefine these 3D keypoints and accurately annotate their corresponding location in the image plane. To address these challenges, we divide the object mesh into several parts $\{\mathcal{P}_1, \mathcal{P}_2, \dots\}$ and select several keypoints for each part $\mathcal{P}_i = \{\mathbf{x}_{3\text{D}}^{(1)}, \mathbf{x}_{3\text{D}}^{(2)}, \dots\}$. In figure 3, we show the keypoints that we selected for annotation. Most keypoints are distributed on the edge of the object mesh. Then the annotators are asked to annotate the position of the keypoints and their corresponding part labels $\mathcal{K} = \{(\mathbf{x}_{2\text{D}}^{(1)}, l_1), (\mathbf{x}_{2\text{D}}^{(2)}, l_2), \dots\}$ for each image.

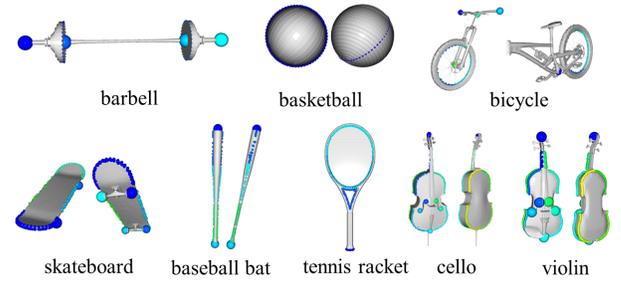


Figure 3: The pre-selected keypoint sets for each object. Different parts are colored using different colors.

Solve the 6D Pose Given the annotations $\{(\mathbf{x}_{2\text{D}}^{(1)}, l_1), (\mathbf{x}_{2\text{D}}^{(2)}, l_2), \dots\}$, the 6D pose of the object is solved by

$$\{\mathbf{R}^*, \mathbf{t}^*\} = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_{(\mathbf{x}_{2\text{D}}, l) \in \mathcal{K}} \min_{\mathbf{x}_{3\text{D}} \in \mathcal{P}_l} \|\Pi(\mathbf{R}\mathbf{x}_{3\text{D}} + \mathbf{t}) - \mathbf{x}_{2\text{D}}\|^2,$$

where Π is the camera perspective projection function. In figure 4, we show several annotation examples. From these examples, we can see that the effectiveness of this annotation scheme.

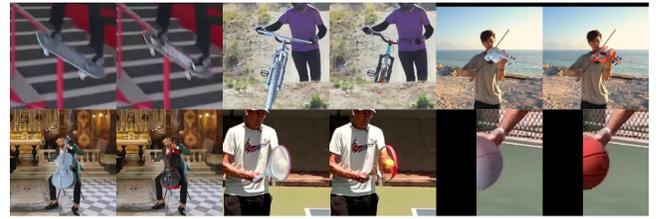


Figure 4: The keypoint annotations and the solved 6D pose.

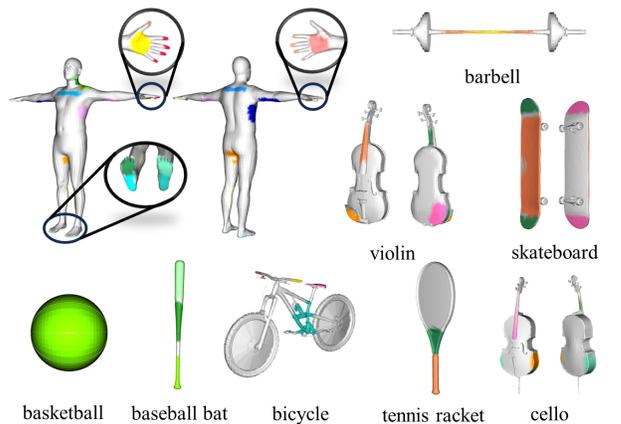


Figure 5: The selected contact regions that are likely under contact during interaction.

Category		barbell	baseball bat	basketball	bicycle	cello	skateboard	tennis bat	violin
Training	Videos	204	372	84	224	204	280	339	184
	Frames	37,869	39,871	36,647	43,094	101,737	101,643	82,820	31,049
Testing	Videos	40	79	22	57	52	63	84	47
	Frames	200	589	130	268	181	511	473	181

Table 1: The scale of the WildHOI dataset.

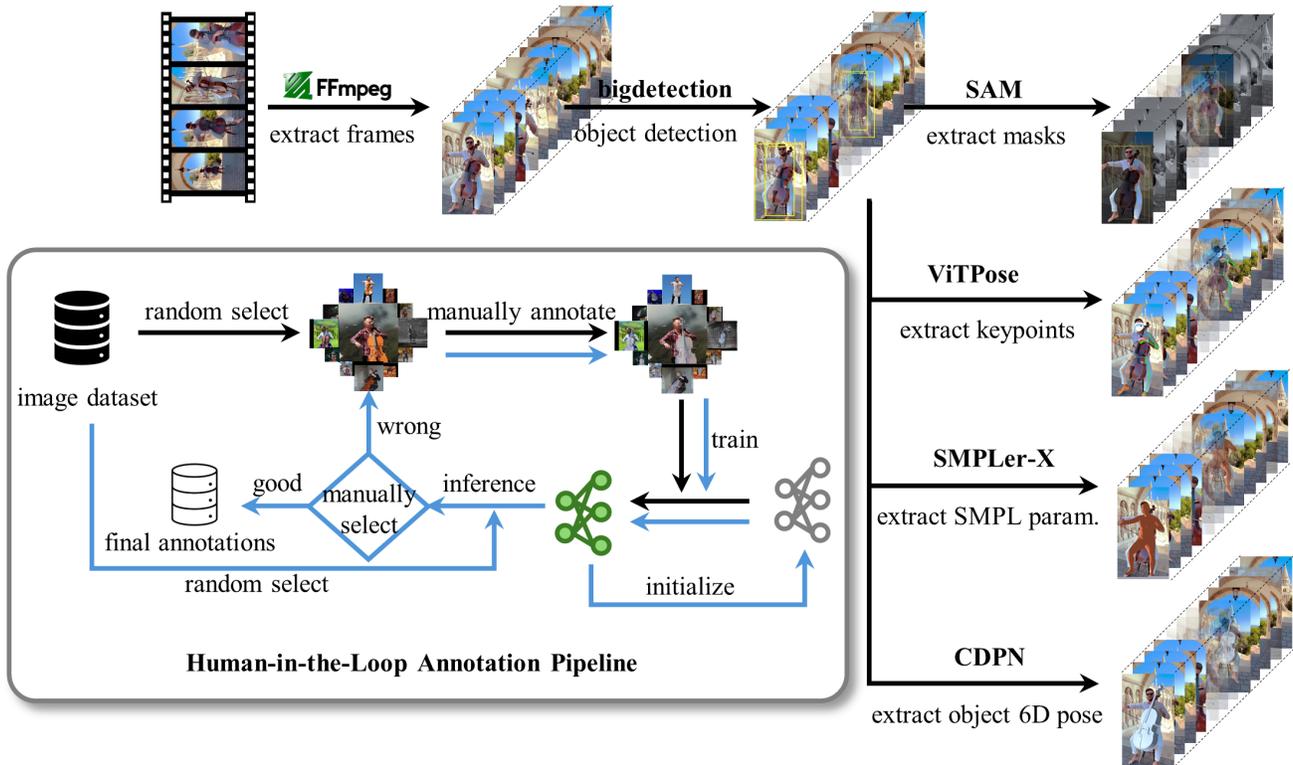


Figure 6: This figure shows the annotation pipeline for the WildHOI dataset. We use state-of-the-art model such as bigdetection, SAM, ViTPose, SMPLer-X, CDPN to extract bounding boxes, masks, wholebody keypoints, SMPL parameters, and object 6D pose for each image. The 6D pose is annotated using a human-in-the-loop annotation pipeline, which involves human annotators validating and correcting the poses predicted by CDPN.

	barbell	baseball	basketball	bicycle	cello	skateboard	tennis	violin	average
PHOSA	3.00	6.96	8.46	1.12	0.55	3.33	5.07	0.00	4.07
Ours	51.50	48.39	30.77	85.07	82.32	45.01	42.07	57.46	52.82
Draw	45.50	44.65	60.77	13.81	17.13	51.66	52.85	42.54	43.11

Table 2: The percentage of better images voted by annotator on WildHOI test dataset.

3.2 Human-Object Spatial Relation Annotation

Contact Annotation For most cases, the relative pose between the human and the object can be tuned through the contact maps. We predefine the possible contact regions on the surface of the SMPL mesh and the object meshes. Then the annotators are asked to annotate which contact region is under contact, rather than annotating the contact points pointwisely. The contact regions we defined are shown in figure 5. With the contact labels, we optimize

the relative pose between the human and the object by minimizing the contact loss. Denote the predefined contact regions for the human as $\{\mathcal{R}_1^h, \mathcal{R}_2^h, \dots\}$ and the contact regions for the object as $\{\mathcal{R}_1^o, \mathcal{R}_2^o, \dots\}$, where each contact region is a set of points. The contact annotation is denoted as $C = \{(i_1, j_1), (i_2, j_2), \dots\}$, where each item represents the pair of the index of contact region in SMPL

mesh and the object mesh respectively.

$$\mathcal{L}_{\text{contact}} = \sum_{(i,j) \in C} \left\{ \frac{1}{|\mathcal{R}_i^h|} \sum_{\mathbf{p}_h \in \mathcal{R}_i^h} \min_{\mathbf{p}_o \in \mathcal{R}_j^o} \|\mathbf{p}_h - \mathbf{p}_o\| + \right. \quad (1)$$

$$\left. \frac{1}{|\mathcal{R}_j^o|} \sum_{\mathbf{p}_o \in \mathcal{R}_j^o} \min_{\mathbf{p}_h \in \mathcal{R}_i^h} \|\mathbf{p}_h - \mathbf{p}_o\| \right\}. \quad (2)$$

The contact loss is optimized with the projection loss and the regularization loss.

Annotation for Non-Contact Interaction Types For non-contact interaction types, we use the interactive tools to manually annotate the 6D pose of the object in the SMPL local coordinate system. As shown in figure 7, the annotators are asked to tune the location and the rotation of the object to make the relative spatial relation between the human and the object seem consistent in other views.

The annotations generated by the above annotation process are visualized in figure 9. We use these pseudo annotations to evaluate the performance of our method on the WildHOI dataset.

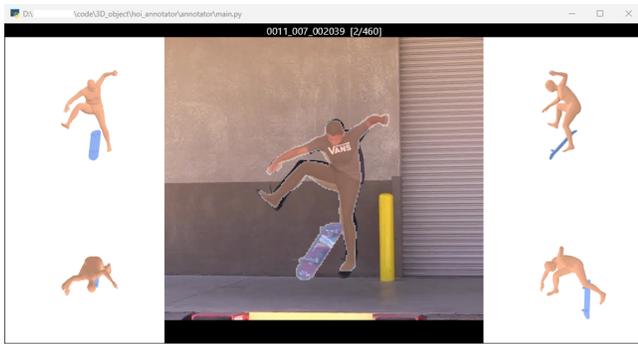


Figure 7: The interactive annotation tool for labeling the pose of the object.

4 ADDITIONAL EXPERIMENTS

Ablation on the Number of Virtual Cameras. In table 3, we show the reconstruction accuracy with varying numbers of the virtual camera m . As the number of virtual cameras increases from 1 to 32, both the SMPL error and the object error decrease. This indicates that using more virtual cameras can lead to more accurate reconstructions. However, using more virtual cameras in the post-optimization stage leads to increased computational complexity with only marginal improvement in accuracy. The optimal choice for the number of virtual cameras is 4-8, striking a balance between accuracy and computational efficiency.

m	1	4	8	16	32
SMPL (cm) ↓	4.77	4.59	4.55	4.54	4.52
Obj. (cm) ↓	13.21	11.55	11.32	11.32	11.23

Table 3: The impact of the numbers of the virtual camera on the reconstruction accuracy.

Human Evaluation on WildHOI-test dataset We also conduct the human evaluation on the WildHOI-test dataset. We render the reconstruction results of different methods onto different viewpoints, shuffle the results, and deliver the results to annotators for evaluation. Annotators are asked to indicate which reconstruction is better. At the same time, if the reconstruction results of both methods are similar or both do not meet the requirements and cannot be distinguished, the annotators should mark them as a draw. The final human evaluation score is computed as the fraction of better images. In table 2, we show the fraction of the better images selected by the annotator. The last line represents a draw, i.e. the annotator cannot distinguish which is better. It can be observed that our method outperforms PHOSA in most categories, especially on the bicycle and cello categories. More qualitative comparisons are shown in figure 10, 11, 12. However, our method fails in some cases, as shown in figure 8.

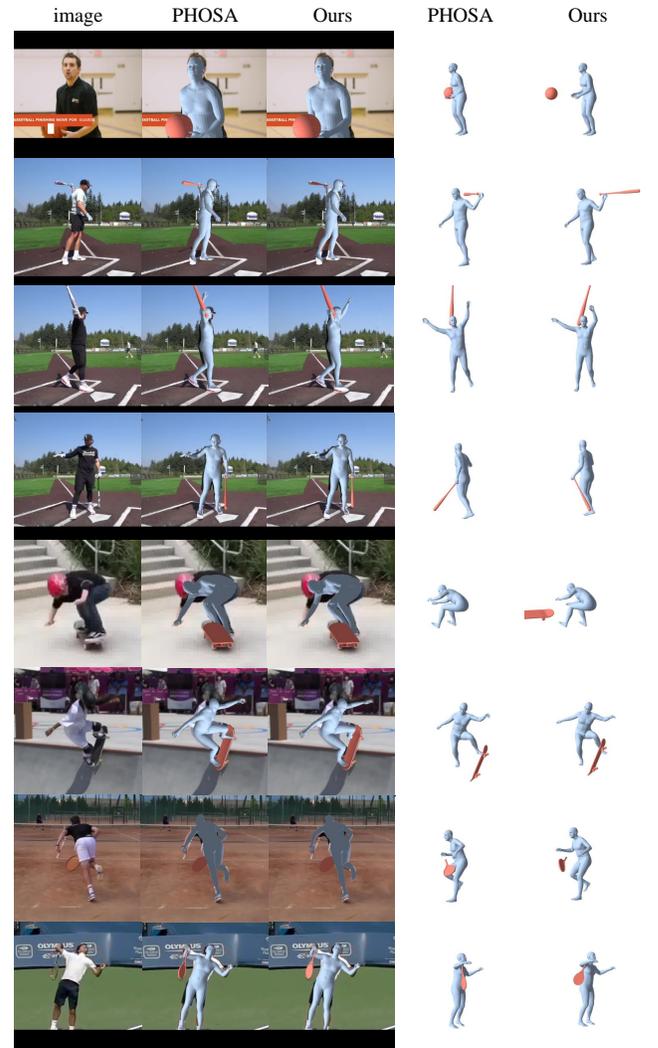


Figure 8: The failures of our method.

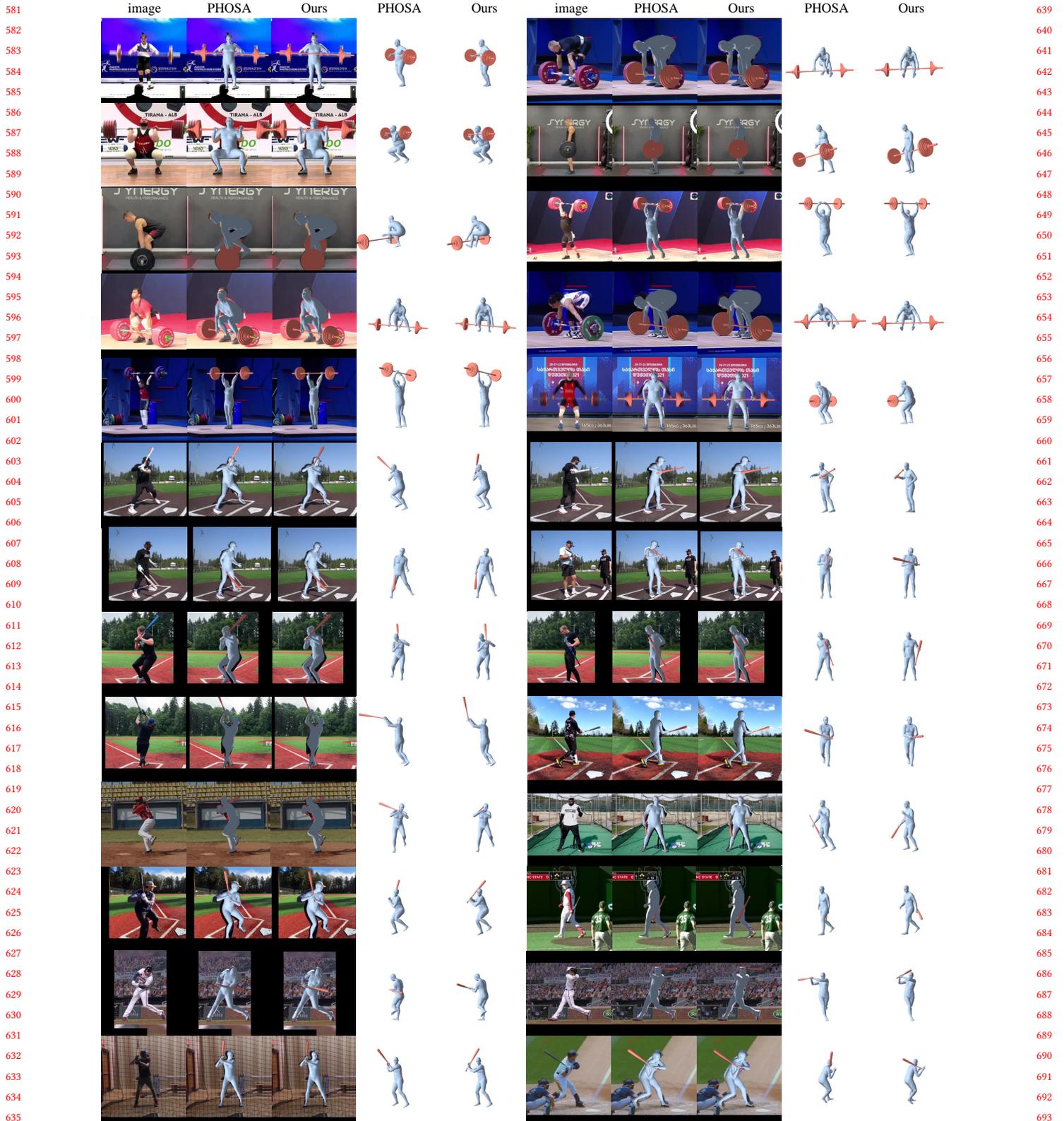


Figure 10: Qualitative Comparison on WildHOI dataset.

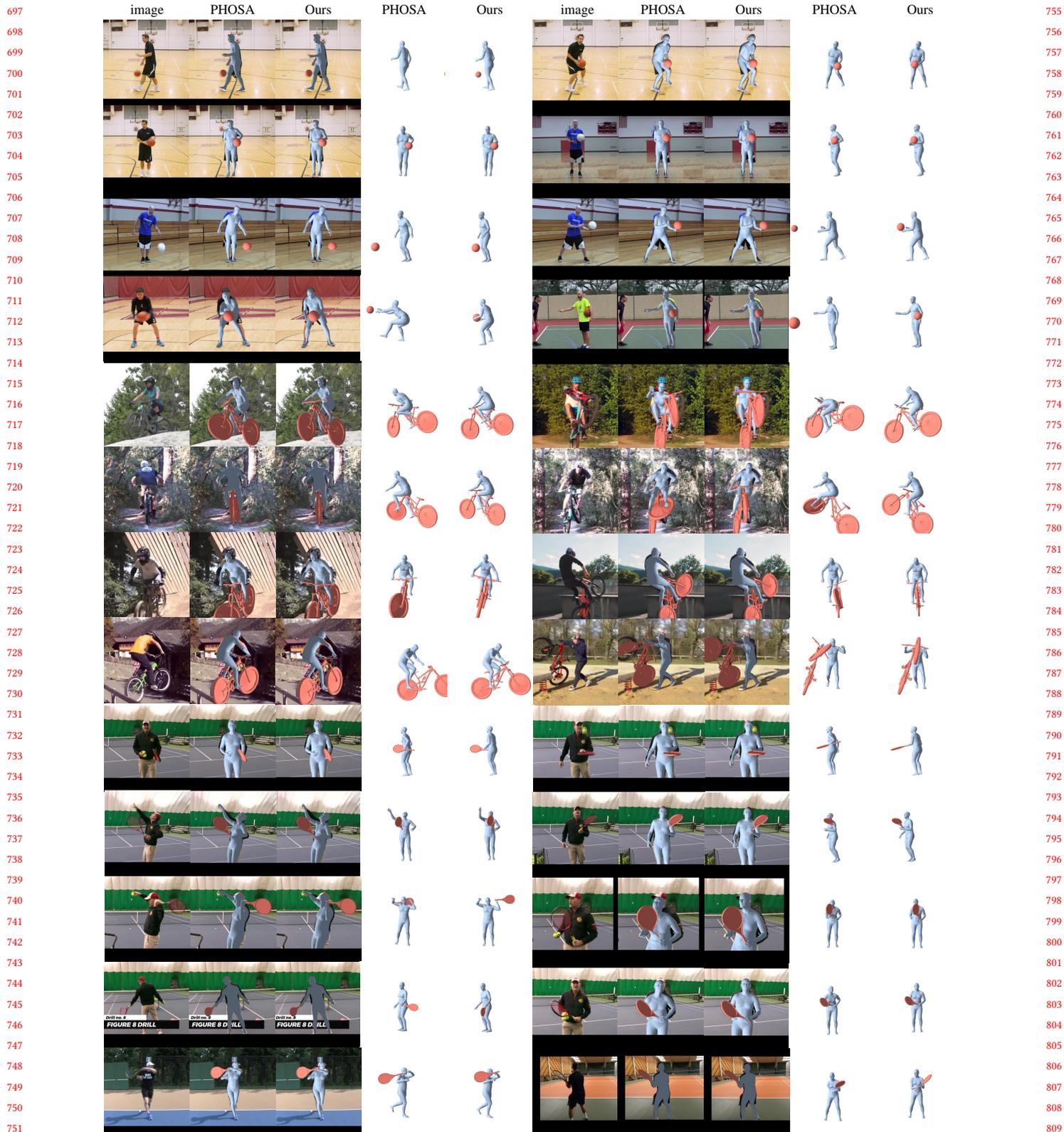


Figure 11: Qualitative Comparison on WildHOI dataset.

