Supplementary: Visibility Aware Human-Object Interaction Tracking from Single RGB Camera

Xianghui Xie Bharat Lal Bhatnagar Gerard Pons-Moll

University of Tübingen, Germany
Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
{xxie, bbhatnag}@mpi-inf.mpg.de, gerard.pons-moll@uni-tuebingen.de

In this supplementary, we first list all the implementation details of our method and then show more ablation study results as well as comparison with CHORE [12] on NTU-RGBD [6] dataset. We end with discussions of failure cases and future works.

1. Implementation details

1.1. Obtaining SMPL-T meshes

To obtain the image-aligned SMPL meshes that have consistent translation (SMPL-T) we keep the SMPL shape parameters and optimize the body pose and global translation values. The loss weights for this optimization are: $\lambda_{\text{J2D}} = 0.09, \lambda_{\text{reg}} = 1.0 \times 10^{-5}, \lambda_a = 25, \lambda_{\text{pi}} = 900$. We optimize the parameters until convergence with a maximum iteration of 1000.

1.2. SIF-Net: SMPL-T conditioned interaction field

A visualization of our SMPL-T triplane rendering and query point projection can be found in Fig. 1. We discuss our network architecture and training details next.

Network architecture. We use the stacked hourglass network [7] for both RGB image encoder f^{enc} and SMPL rendering encoder f^{tri}. We use 3 stacks for f^{tri} and the output feature dimension is $d_{\text{o}}^{\text{tri}} = 64$. Hence $f^{\text{tri}} : \mathbb{R}^{H\times W} \mapsto \mathbb{R}^{H/4\times W/4\times 64}$ where $H = W = 512$. We also use 3 stacks for f^{enc} but the feature dimension is $d_{\text{o}}^{\text{enc}} = 256$. Hence $f^{\text{enc}} : \mathbb{R}^{H\times W\times 5} \mapsto \mathbb{R}^{H/4\times W/4\times 256}$. We also concatenate the image features extracted from the first convolution layer and query point coordinate to the features. Thus the total feature dimension to our decoders is: $d = (d_1^{\text{tri}} + d_{\text{o}}^{\text{tri}}) \times 3 + d_1^{\text{enc}} + d_{\text{o}}^{\text{enc}} + 3 = 611$, here $d_1^{\text{tri}} = 32, d_1^{\text{enc}} = 64$. All decoders consist of three FC layers with ReLU activation and one output FC layer with hidden dimension of 128 for the intermediate features. The visibility decoder f^v additionally has a sigmoid output activation layer. The output shape is $2, 14, 9, 3, 1$ for f^u, f^p, f^R, f^c, f^v respectively.

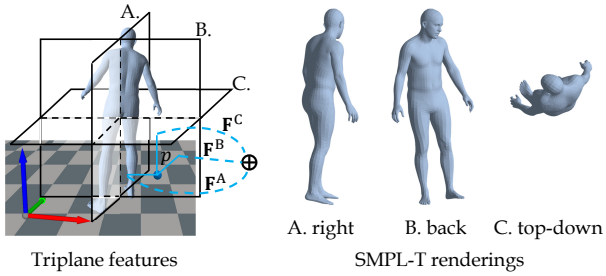


Figure 1. Visualization of our SMPL-T triplane feature extraction and rendering. The triplane origin is placed at the SMPL-T body center and we render the mesh from three views using orthographic projection: righ-left (A), back-front (B) and top-down (C). The query point p is projected into the three planes using same projection for rendering and we extract pixel aligned features $\mathbf{F}^A, \mathbf{F}^B, \mathbf{F}^C$ from the feature planes respectively. Note that we render the SMPL-T with color here for visualization, the actual input to our network are silhouette images only.

Training. All feature encoders and decoders are trained end to end with the loss: $L = \lambda_u(L_{u_h} + L_{u_o}) + \lambda_p L_p + \lambda_R L_R + \lambda_c L_c + \lambda_v L_v$. Here L_{u_i} is the L_1 distance between ground truth and predicted unsigned distance to human or object surface [12]. L_p is a standard categorical cross entropy loss for SMPL part correspondence prediction. L_R, L_c, L_v are mean square losses between ground truth and predicted values for rotation matrix, translation vector and visibility score respectively. The loss weights are: $\lambda_u = 1.0, \lambda_R = 0.006, \lambda_c = 500, \lambda_c = \lambda_v = 1000$. The model is trained for 18 epochs and it takes 25h to converge on a machine with 4 RTX8000 GPUs each with 48GB memory. The training batch size is 8.

1.3. HVOP-Net: object pose under occlusion

We use three transformers $f^s, f^o, f^{\text{comb}}$ to aggregate features from SMPL-T, object pose and joint human object information respectively. We use the 6D vector [19] to repre-

sent the ration matrix of SMPL-T and object pose parameters. Hence the SMPL-T pose dimension is $24 \times 6 + 3 = 147$, where 3 denotes the global translation. We predict the object rotation only thus the object data dimension is 6. The SMPL-T transformer f^s consists of an MLP: $\mathbb{R}^{T\times 147} \mapsto \mathbb{R}^{T\times 128}$ and two layers of multi-head self-attention (MHSA) module [9] with 4 heads. Similarly, the object transformer f^o consists of an MLP: $\mathbb{R}^{T\times 6} \mapsto \mathbb{R}^{T\times 32}$ and two layers of MHSA module with 2 heads. The joint transformer f^{comb} consists of 4 layers of MHSA module with 1 head only. GeLU activation is used in all MHSA modules. We finally predict the object pose using two MLP layers with an intermediate feature dimension of 32 and LeakyReLU activation.

The model is trained to minimize the L_1 losses of pose value and accelerations: $L = \lambda_{\text{pose}} L_{\text{pose}} + \lambda_{\text{accel}} L_{\text{accel}}$, where $\lambda_{\text{pose}} = 1.0, \lambda_{\text{accel}} = 0.1$. It is trained on a server with 2 RTX8000 GPUs, each GPU has 48GB memory capacity. It takes around 7h to converge (64 epochs).

1.4. SmoothNet for SMPL-T and object

We use SmoothNet [17] to smooth our SMPL-T and SIF-Net object pose predictions. We use exactly the same model and training strategy proposed by the original paper. The input to the SMPL-T SmoothNet is our estimated SMPL-T pose and translation (relative to the first frame). The input to the object SmoothNet is the object rotation (6D vector). Following the standard practice of SmoothNet [17], we train both models on the predictions from the BEHAVE [1] training set. Note that we do not fine-tune them on InterCap [3] dataset. We evaluate this component in Sec. 2.3.

1.5. Visibility aware joint optimization

The objective function defined in Eq. 2 is highly non-convex thus we solve this optimization problem in two stages. We first optimize the SMPL pose and shape parameters using human data term only. We then optimize the object parameters using the object and contact data terms. The loss weights are set to: $\lambda_{\text{reg}} = 2.5 \times 10^{-4}, \lambda_{\text{ah}} = 10^4, \lambda_h = 10^4, \lambda_p = t \times 10^{-4}, \lambda_o = 900, \lambda_{\text{occ}} = 9 \times 10^{-4}, \lambda_{\text{ao}} = 225, \lambda_c = 900$, where λ_c is the loss weight for the contact data term defined in Eq. 5.

2. Additional ablation results

2.1. Further evaluation of SMPL-T conditioning

We show some example images from one sequence in Fig. 2 to evaluate the importance of our SMPL-T conditioning. It can be seen that without this conditioning, the human is reconstructed at fixed depth, leading to inconsistent relative translation across time. Our method predicts more coherent relative human translation and more accurate object pose.

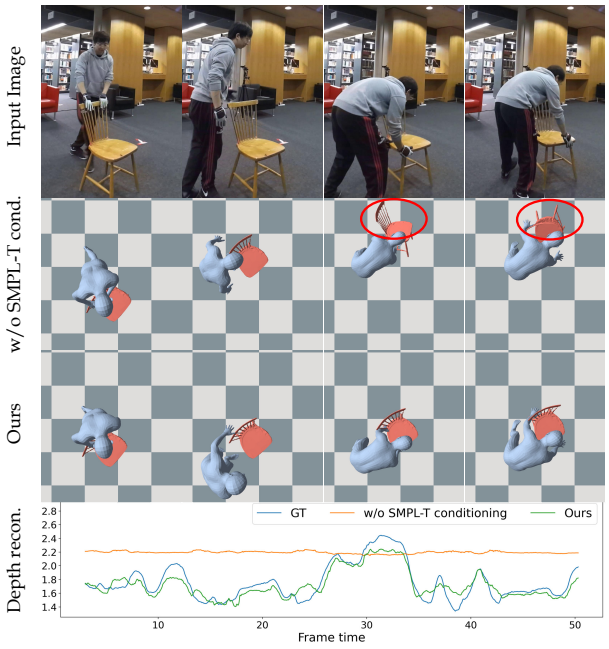


Figure 2. Evaluating SMPL-T conditioning for neural field prediction. We can see that without conditioning on SMPL-T meshes, the object pose prediction is worse and human is reconstructed at fixed depth, leading to inconsistent relative location across frames. Our method recovers the relative translation more faithfully and obtain better object pose predictions.

To further evaluate SMPL-T conditioning, we compute the object pose error from the raw network predictions and compare it with the object pose of CHORE which is also the raw prediction from the network. The pose error is computed as Chamfer distance (CD) and vertex to vertex (v2v) error after centring the prediction and GT mesh at origin. We also report the translation error (transl.) as the distance between predicted and GT translation. The results are shown in Tab. 1. We can clearly see that our SMPL feature improves both the raw object pose prediction and distance fields (results after optimization are also improved).

Method	Raw prediction			After opt. w=10	
	CD↓	v2v↓	transl.↓	SMPL↓	obj.↓
w/o SMPL-T	5.56	16.10	14.28	14.40	17.29
Ours	**3.98**	**12.34**	**9.53**	**8.03**	**8.23**

Table 1. Importance of SMPL-T conditioning (errors in cm). We can see that our SMPL-T feature improves both the raw object pose prediction and distance fields (after opt.). Without our SMPL-T conditioning, the reconstructed translation is not consistent across frames, leading to large errors after alignment of temporal window of 10s (w=10).

2.2. Comparing different pose prediction methods

We show some example comparisons of different object pose prediction methods under heavy occlusions in Fig. 4.

Method	Chamfer	v2v	Acceleration
w/o SmoothNet	8.71	9.84	1.38
w/ SmoothNet	**8.01**	**9.12**	**1.18**

Table 2. Ablate SMPL SmoothNet (errors in cm). We can see that SmoothNet [17] improves the overall smoothness and slightly reduces the pose errors.

Method	Chamfer	v2v	Translation
a. Raw prediction	5.03	10.39	10.01
b. Raw + SmoothNet	4.22	8.60	10.16
c. Raw + our pose pred.	4.09	8.02	10.20
d. Our full model	**3.62**	**7.20**	**9.96**

Table 3. Ablate SmoothNet for object pose prediction (errors in cm). We can see our pose prediction (c) is better than SmoothNet [17] (b). Combing both we obtain the best result (d).

We compare our method against: 1). Raw prediction from our SIF-Net. 2). Linearly interpolate the occluded poses from visible frames (SLERP). 3). CMIB [4], a transformer based model trained to infill the object motion using visible frames. Note here the evaluation is based on the final tracking results and we report the object errors only as the difference of SMPL error is very small. Similar to Sec. 2.1, the object errors are computed as Chamfer distance, v2v error and translation error.

It can be seen that the raw pose prediction is noisy due to occlusion. SLERP and CMIB corrects some pose errors but is not robust as they do not leverage the human information. Our method is more accurate as it takes the human context and object pose into account.

2.3. Evaluating SmoothNet

SmoothNet [17] is used to smooth the SMPL-T parameters after 2D keypoint based optimization. We evaluate this step by computing the SMPL errors, shown in Tab. 2. We can see that SmoothNet reduces the SMPL error slightly.

We also use SmoothNet to smooth the object pose before sending it to our human and visibility aware object pose prediction network. SmoothNet cannot correct errors under long-term occlusions. However, it provides smoother object motion for visible frames which can benefit our pose prediction network. We evaluate this using object pose errors and report the results in Tab. 3. It can be seen that our method (Tab. 3c) works better than SmoothNet (Tab. 3b) on raw predictions. Nevertheless, with smoothed pose after SmoothNet, our method achieves the best result (Tab. 3 d).

2.4. Runtime cost

SMPL-T pre-fitting and joint optimization can be run in batches hence the average runtime per frame is not long: SMPL-T pre-fitting: 6.38s, SIF-Net object pose prediction: 0.89s, HVOP-Net: 1.3ms, joint optimization: 9.26s, total: 16.53s. Compared to CHORE ($\sim$12s/frame) [12], the additional cost is mainly from the SMPL-T pre-fitting. Yet, SMPL-T conditioning allows faster convergence of joint optimization and much better reconstruction. Since we use efficient 2D encoder instead of 3D encoder, it takes only 1.05GB GPU memory to load the SIF-Net model. This allows us to do joint optimization with batch size up to 128 on a GPU with 48GB memory.

3. Generalization to NTU-RGBD dataset

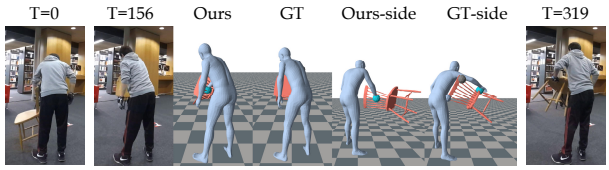
Obtaining input masks. Unlike BEHAVE and InterCap where the human and object masks are provided by the dataset, there are no masks in NTU-RGBD. To this end, we run DetectronV2 [11] to obtain the human masks. We manually segment the object in the first frame using interactive segmentation [8] ($<$1min/image) and then use video segmentation [2] to propagate the masks. The overhead of 1min/video manual label is small.

We show more results from our method on NTU-RGBD dataset [6] and compare against CHORE [12] in Fig. 5. It can be seen that CHORE may predict some reasonable object pose but it fails quite often to capture the fine-grained contacts between the human and object. Our method obtains more coherent reconstruction for different subjects, human-backpack interactions, camera view points and backgrounds. Please see our project website for comparison in full sequences.

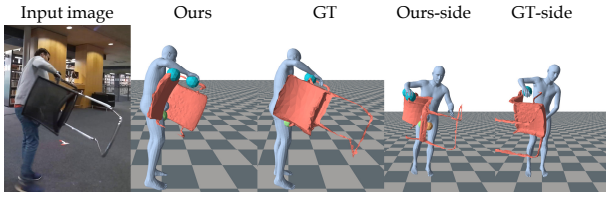
4. Limitations and future works

Although our method works robustly under heavy occlusions, there are still some limitations. Firstly, we assume known object templates for tracking, an interesting direction is to build such a template from videos as demonstrated by recent works [10, 14–16]. Secondly, it would be interesting to model multi-person or even multi-object interactions which is a more realistic setting in real-life applications. In addition, the backpack can also deform non-rigidly which is not modelled in our method. Further works can incorporate the surface deformation [5] or object articulation [13] into the human object interaction. We leave these for future works.

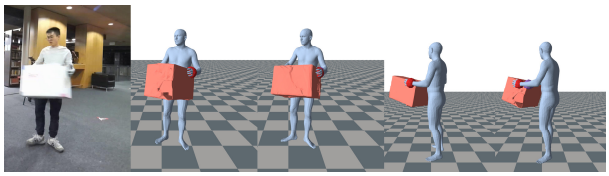
We identify three typical failure cases of our method, some examples are shown in Fig. 3. The first typical failure case comes from heavy occlusion when the object undergoes significant changes (object pose and contact locations) between two visible frames. In this case, it is very difficult to track the pose and contact changes accurately (Fig. 3 A). Second typical failure is due to the difficulty of pose prediction itself even the object is fully visible. In this case the object pose is uncommon and the network failed to predict it correctly (Fig. 3 B). Another failure is caused by symmetric objects. Our optimization minimizes the 2D mask loss and contact constraints but the network is confused by the symmetry and the initial pose prediction is not semantically



A. Significant object pose change between two visible frames.



B. Uncommon object pose.



C. Symmetric object.

Figure 3. Failure cases analysis. We show three typical failure cases of our method: A. The occluded object pose (T=156) changes significantly between two visible frames (T=0 and T=319) and it is difficult to accurate track the contact changes. B. The object pose is not commonly seen during interaction and it is difficult to predict for this rare pose. C. The object is symmetric. The joint optimization satisfies the object mask and contacts but is not semantically correct.

correct (Fig. 3 C). In addition, the training data for these objects is very limited (only $1/3$ of other objects). More training data or explicitly reasoning about the symmetry [18] can be helpful.

Figure 4. Comparing different object pose prediction method under heavy occlusions. Raw prediction is from our SIF-Net output, SLERP denotes linear interpolation and CMIB is from [4]. We can see SLERP and CMIB can correct some errors (row 5) but they do not take the human motion into account hence often fail in more challenging cases. Our method is more robust as it leverages information from both human motion and object pose from visible frames.

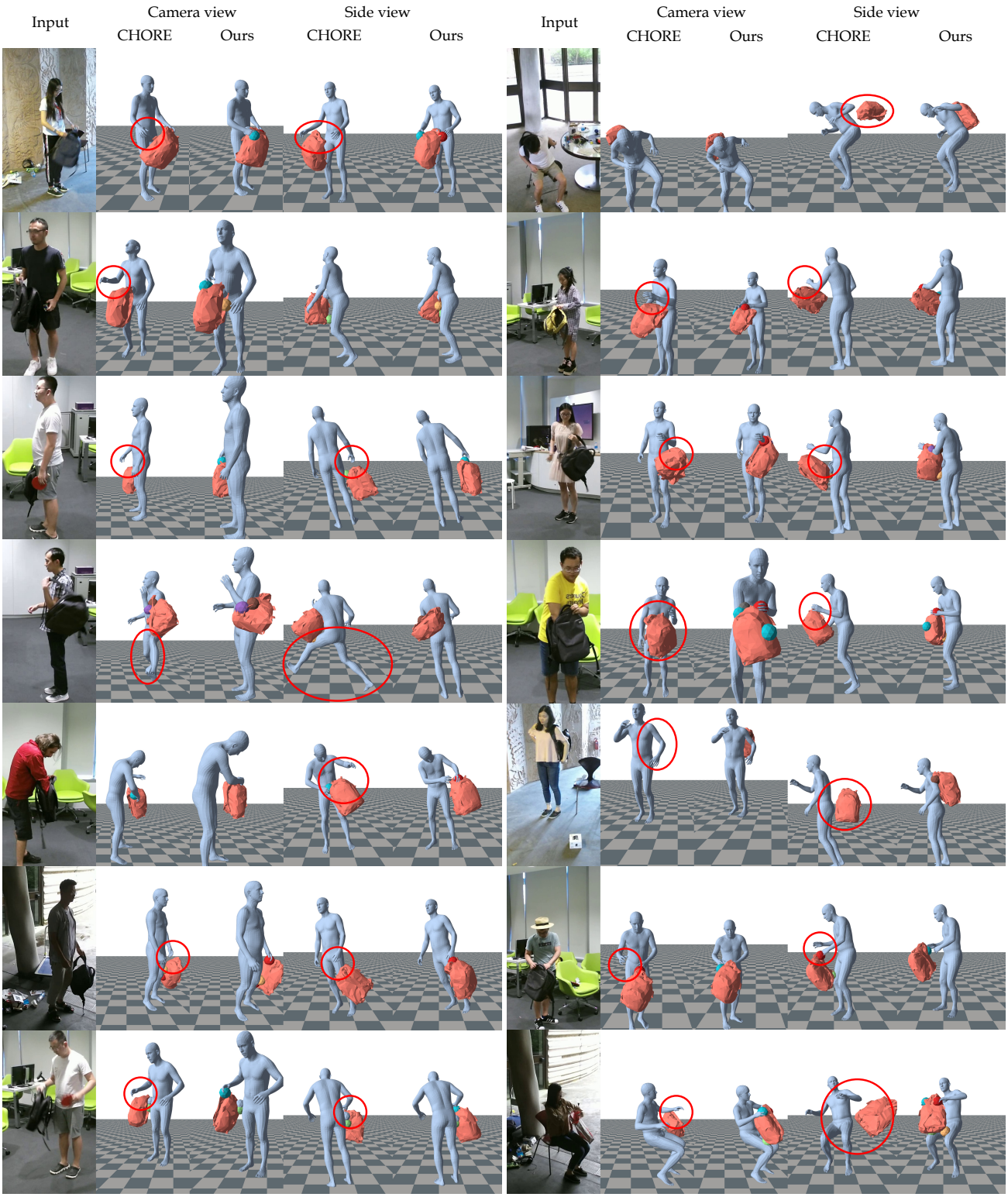


Figure 5. Comparing our method with CHORE [12] on NTU-RGBD [6] dataset. It can be seen that CHORE does not capture the realistic contacts between the person and the backpack. Our method recovers the 3D human, the object and contacts more faithfully in different interaction types, camera view points and backgrounds.

References

[1] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2

[2] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. In *CVPR*, 2021. 3

[3] Yinghao Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. InterCap: Joint markerless 3D tracking of humans and objects in interaction. In *German Conference on Pattern Recognition (GCPR)*, volume 13485 of *Lecture Notes in Computer Science*, pages 281–299. Springer, 2022. 2

[4] Jihoon Kim, Taehyun Byun, Seungyoun Shin, Jungdam Won, and Sungjoon Choi. Conditional motion inbetweening. *Pattern Recognition*, page 108894, 2022. 3, 5

[5] Zhi Li, Soshi Shimada, Bernt Schiele, Christian Theobalt, and Vladislav Golyanik. Mocapdeform: Monocular 3d human motion capture in deformable scenes. In *International Conference on 3D Vision (3DV)*, 2022. 3

[6] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 3, 6

[7] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. 1

[8] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020. 3

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[10] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. DOVE: Learning deformable 3d objects by watching videos. *arXiv preprint arXiv:2107.10844*, 2021. 3

[11] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

[12] Xianghui Xie, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Chore: Contact, human and object reconstruction from a single rgb image. In *European Conference on Computer Vision (ECCV)*. Springer, October 2022. 1, 3, 6

[13] Xiang Xu, Hanbyul Joo, Greg Mori, and Manolis Savva. D3d-hoi: Dynamic 3d human-object interactions from videos. *arXiv preprint arXiv:2108.08420*, 2021. 3

[14] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *CVPR*, 2021. 3

[15] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In *NeurIPS*, 2021. 3

[16] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *CVPR*, 2022. 3

[17] Ailing Zeng, Lei Yang, Xuan Ju, Jiefeng Li, Jianyi Wang, and Qiang Xu. Smoothnet: A plug-and-play network for refining human poses in videos. In *European Conference on Computer Vision*. Springer, 2022. 2, 3

[18] Linfang Zheng, Aleš Leonardis, Tze Ho Elden Tse, Nora Horanyi, Hua Chen, Wei Zhang, and Hyung Jin Chang. TP-AE: Temporally Primed 6D Object Pose Tracking with Auto-Encoders. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10616–10623, Philadelphia, PA, USA, May 2022. IEEE Press. 4

[19] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1