Abstract

Capturing the interactions between humans and their environment in 3D is important for many applications in robotics, graphics, and vision. Recent works to reconstruct the 3D human and object from a single RGB image do not have consistent relative translation across frames because they assume a fixed depth. Moreover, their performance drops significantly when the object is occluded. In this work, we propose a novel method to track the 3D human, object, contacts between them, and their relative translation across frames from a single RGB camera, while being robust to heavy occlusions. Our method is built on two key insights. First, we condition our neural field reconstructions for human and object on per-frame SMPL model estimates obtained by pre-fitting SMPL to a video sequence. This improves neural reconstruction accuracy and produces coherent relative translation across frames. Second, human and object motion from visible frames provides valuable information to infer the occluded object. We propose a novel transformer-based neural network that explicitly uses object visibility and human motion to leverage neighbouring frames to make predictions for the occluded frames. Building on these insights, our method is able to track both human and object robustly even under occlusions. Experiments on two datasets show that our method significantly improves over the state-of-the-art methods. Our code and pretrained models are available at: https://virtualhumans.mpi-inf.mpg.de/VisTracker.

1. Introduction

Perceiving and understanding human as well as their interaction with the surroundings has lots of applications in robotics, gaming, animation and virtual reality etc. Accurate interaction capture is however very hard. Early works employ high-end systems such as dense camera arrays [8, 17, 37] that allow accurate capture but are expensive to deploy. Recent works [6, 34, 36] reduce the requirement to multi-view RGBD cameras but it is still complicated to setup the full capture system hence is not friendly for consumer-level usage. This calls for methods that can capture human-object interaction from a single RGB camera, which is more convenient and user-friendly.

However, reasoning about the 3D human and object from monocular RGB images is very challenging. The lack of depth information makes the predictions susceptible to depth-scale ambiguity, leading to temporally incoherent tracking. Furthermore, the object or human can get heavily occluded, making inference very hard. Prior work PHOSA [91] relies on hand-crafted heuristics to reduce the ambiguity but such heuristic-based method is neither very accurate nor scalable. More recently, CHORE [80] combines neural field reconstructions with model based fitting obtaining promising results. However, CHORE, assumes humans are at a fixed depth from the camera and predicts scale alone, thereby losing the important relative translation across frames. Another limitation of CHORE is that it is not robust under occlusions as little information is available from single-frame when the object is barely visible. Hence CHORE often fails in these cases, see Fig. 3.

In this work, we propose the first method that can track
both human and object accurately from monocular RGB videos. Our approach combines neural field predictions and model fitting, which has been consistently shown to be more effective than directly regressing pose \cite{4–6, 80}. In contrast to existing neural field based reconstruction methods \cite{61, 80}, we can do tracking including inference of relative translation. Instead of assuming a fixed depth, we condition the neural field reconstructions (for object and human) on per frame SMPL estimates (SMPL-T) including translation in camera space obtained by pre-fitting SMPL to the video sequence. This results in coherent translation and improved neural reconstruction. In addition, we argue that during human-object interaction, the object motion is highly correlated with the human motion, which provides us valuable information to recover the object pose even when it is occluded (see Fig. 1 column 3-4). To this end, we propose a novel transformer based network that leverages the human motion and object motion from nearby visible frames to predict the object pose under heavy occlusions.

We evaluate our method on the BEHAVE \cite{6} and InterCap dataset \cite{35}. Experiments show that our method can robustly track human, object and realistic contacts between them even under heavy occlusions and significantly outperforms the current state of the art method, CHORE \cite{80}. We further ablate the proposed SMPL-T conditioning and human and visibility aware object pose prediction network and demonstrate that they are key for accurate human-object interaction tracking.

In summary, our key contributions include:

- We propose the first method that can jointly track full-body human interacting with a movable object from a monocular RGB camera.
- We propose SMPL-T conditioned interaction fields, predicted by a neural network that allows consistent 4D tracking of human and object.
- We introduce a novel human and visibility aware object pose prediction network along with an object visibility prediction network that can recover object poses even under heavy occlusions.
- Our code and pretrained models are publicly available to foster future research in this direction.

2. Related Work

In this section, we first review recent works that deal with human or object pose estimation and tracking separately. We then discuss recent progresses that model human and object interactions and works that deal with occlusions explicitly.

Human or object pose estimation and tracking. After the introduction of SMPL \cite{49} body model, tremendous progress has been made in human mesh recovery (HMR) from single images \cite{2, 3, 7, 18, 43, 52, 55} or videos \cite{20, 40, 56, 57, 89}. We refer readers to a recent review of HMR methods in \cite{70}. On the other hand, deep learning method has also significantly improved object 6D pose estimation from single RGB images \cite{22, 25, 33, 45, 50, 53, 73}. However, object pose tracking has received less attention and most works focus on RGBD inputs \cite{21, 65, 75, 76, 94}. Two works explore the camera localization ideas from SLAM communities and can track object from RGB videos \cite{48, 68}. Nevertheless, they heavily rely on visual evidence and the performance is unknown under heavy occlusions. They also do not track human-object interactions.

Human-object interaction. Modelling human object interaction is an emerging research topic in recent years. Hand-object interaction is studied with works modelling hand-object interaction from RGB \cite{19, 24, 30, 38, 85}, RGBD \cite{9, 11, 28} or 3D inputs \cite{10, 54, 69, 95}. There are also works that model human interacting with a static scene \cite{27, 29, 34, 63, 77, 86, 87} or deformable surface \cite{46}. More recently, the release of BEHAVE \cite{6} and InterCap \cite{35} datasets allows bench-marking of full-body interacting with a movable object. However, human-object interaction capture usually deploys multi-view RGB \cite{67} or RGBD \cite{6, 9, 20, 28, 28, 34, 35, 93} cameras. Only a few works \cite{74, 80, 91} reconstruct dynamic human and object from monocular RGB input and our experiments show that they are not suitable for tracking.

Pose estimation under occlusion. Most existing methods assume occlusion-free input images hence are not robust under occlusions. Only a few methods address human pose estimation under partial occlusions \cite{26, 41, 42, 58, 59, 92} or long term occlusions \cite{89}. For object pose estimation, pixel-wise voting \cite{53} and self-occlusion \cite{22} are explored for more robust prediction under occlusions. More recently, TP-AE \cite{94} predicts object occlusion ratio to guide pose estimation but relies on depth input. Although being impressive on separate human or object pose estimation, these methods do not reason about human-object interaction. Our method is the first one that takes both human and object visibility into account for interaction tracking.

3. Method

We present a novel method for jointly tracking the human, the object and the contacts between them, in 3D, from a monocular RGB video. The first main challenge in monocular tracking is the estimation of human and object translations in camera space due to the depth-scale ambiguity problem. Existing method, CHORE \cite{80}, reconstructs human and object at a fixed depth to the camera, leading to inconsistent 3D translation across frames. Our key idea is to fit a SMPL model with single shape parameters to a video sequence to obtain consistent relative translation across frames. We call the estimated SMPL as SMPL-T
Figure 2. Given an input RGB sequence of a human interacting with an object and their corresponding human-object masks (A), we aim to reconstruct and track the 3D human, object, and the contacts between them (E). Our first key idea is a SMPL-T conditioned interaction field network (SIF-Net, details in Sec. 3.3) that predicts neural fields conditioned on estimated SMPL meshes in camera space (col. B, SMPL-T, details in Sec. 3.2). SMPL-T conditioning provides us temporally consistent relative translation, which is important for coherent 4D tracking. Our second key insight is to predict object pose under occlusions (D) leveraging human motion and object visibility information (HVOP-Net, details in Sec. 3.4). This prediction provides robust object tracking for frames with heavy occlusion. We then jointly optimize human and object (details in Sec. 3.5) to satisfy image observations and contact constraints.

and describe it in more details in Sec. 3.2. Based on the estimated SMPL-T, we then jointly model the 3D human, object and the interactions using our proposed SMPL-T conditioned Interaction Fields network (SIF-Net, Sec. 3.3). Object tracking from a single-frame only, is difficult when the object is barely visible. Hence we introduce a Human and Visibility aware Object Pose Network (HVOP-Net) that leverages human and object motion from visible frames to recover the object under occlusion (Sec. 3.4). We then use the SIF-Net and HVOP-Net outputs to optimize SMPL model and object pose parameters of this sequence to satisfy neural prediction and image observations (Sec. 3.5). An overview of our approach can be found in Fig. 2.

3.1. Preliminaries

In this work, we focus on a single human interacting with an object, which is a common setting in other hand-object interaction [69, 85, 95] and full body-object interaction works [6, 35, 80]. We represent human using the SMPL [49] body model \( H(\theta, \beta) \) that parameterises the 3D human mesh using pose \( \theta \) (including global translation) and shape \( \beta \) parameters. The object is represented by a known mesh template and we estimate the rotation \( R^o \in SO(3) \) and translation \( t^o \in \mathbb{R}^3 \) parameters. Given a video sequence \( \{I_1, ..., I_T\} \) where \( I_j \in \mathbb{R}^{H \times W \times 5} \) (RGB, human and object masks), our goal is to estimate the SMPL shape \( \beta \), a sequence of SMPL pose \( \Theta = \{\theta_1, ..., \theta_T\} \), object rotation \( R^o = \{R^o_1, ..., R^o_T\} \) and translation \( T^o = \{t^o_1, ..., t^o_T\} \) parameters that satisfy 2D image observation and realistic interaction constraints between the human and the object.

3.2. SMPL-T: Temporally consistent SMPL meshes in camera space

Our first step is to obtain SMPL meshes in camera space that have consistent translation in a video sequence. We leverage 2D body keypoint predictions from openpose [12] and natural motion smoothness for this. Specifically, we use FrankMocap [60] to initialise SMPL-T pose \( \Theta = \{\theta_1, ..., \theta_T\} \) and shape \( \beta = \{\beta_1, ..., \beta_T\} \) parameters for a sequence of images. Note here the original SMPL meshes from FrankMocap are centered at origin. We average the SMPL-T shape parameters over the sequence as the shape for the person and optimize the SMPL-T global translation and body poses to minimize 2D reprojection and temporal smoothness error:

\[
E(\Theta) = \lambda_{12D} L_{12D} + \lambda_{reg} L_{reg} + \lambda_a L_{accel} + \lambda_{pi} L_{pi}
\]

where \( L_{12D} \) is the sum of body keypoint reprojection losses [7] over all frames and \( L_{reg} \) is a regularization on body poses using priors learned from data [49, 52, 71]. \( L_{accel} \) is a temporal smoothness term that penalizes large accelerations over SMPL-T vertices \( H_i; L_{accel} = \sum_{i=0}^{T-2} ||H_i - 2H_{i+1} + H_{i+2}||^2_2 \). \( L_{pi} \) is an L2 loss between optimized and initial body...
pose, prevents the pose from deviating too much from initialization. \( \lambda \) denotes the loss weights detailed in Supp.

Note that this optimization does not guarantee that we get the absolute translation in the world coordinates, it just ensures that our predictions will be consistent with the SMPL model over a sequence, i.e. we will be off by one rigid transformation for the entire sequence.

### 3.3. SIF-Net: SMPL-T conditioned interaction field

Our SMPL-T provides translation about the human but does not reason about the object and the interaction between them. Existing method CHORE [80] can jointly reason human and object and interaction while having consistent relative translation. We model this using a single neural network which we call SIF-Net. The input to SIF-Net consists of RGB image, human and object masks, and our estimated SMPL-Meshes. With features from SMPL-Meshes and input images, it then predicts interaction field which consists of human and object distance fields, SMPL-Mesh part correspondence field, object pose and visibility field.

**SIF-Net feature encoding.** Existing neural implicit methods [61, 62, 80] rely mainly on features from input image, a main reason that limits their human prediction at fixed depth. Instead, we extract features from both our estimated SMPL-Meshes and input image, providing more distinct features for the query points along the same ray. Inspired by EG3D [13], we use the triplane representation for SMPL-Mesh feature learning due to its efficiency. Specifically, we use orthographic camera \( \pi^o(\cdot) \), to render the SMPL-Mesh silhouette from right, back and top-down views and obtain three images \( S_r^i, S_b^i, S_t^i \) respectively, where \( S_j^i \in \mathbb{R}^{H \times W} \), see supplementary for more visualization. Note here the triplane origin is placed at our SMPL-Mesh center (Fig. 2 B). We then train an image encoder \( f^m(\cdot) \) that extracts a pixel aligned feature grid \( D_j^i \in \mathbb{R}^{H_j \times W_j \times C_j} \) from each rendered view \( S_j^i \), where \( j \in \{r, b, t\} \) and \( H_j, W_j, C_j \) are the feature grid dimensions. To extract features for a query point \( p \in \mathbb{R}^3 \), we project \( p \) into the three planes using the same orthographic projection \( \pi^o_p = \pi^o(p) \) and extract local features \( D_p^i = (D_r^i(\pi^o_p), D_b^i(\pi^o_p), D_t^i(\pi^o_p)) \) using bilinear interpolation.

In addition to the SMPL-Mesh features, SIF-Net also extracts information from input images. More specifically, we train an image encoder \( f^{enc}(\cdot) \) to extract feature grid \( Z_i \in \mathbb{R}^{H_f \times W_f \times C_f} \) from input image \( I_i \in \mathbb{R}^{H \times W \times 5} \), here \( H_f, W_f, C_f \) and \( H, W \) are feature grid and input image dimensions respectively. Given query point \( p \in \mathbb{R}^3 \), we project it to 2D image using full perspective projection \( \pi_p = \pi(p) \) and extract pixel-aligned features \( Z_p^i = Z_i(\pi_p) \). The input image feature is concatenated with the SMPL-Mesh feature to form an input and translation aware point feature: \( F_p^i = (Z_p^i, D_p^i) \).

**SIF-Net predictions.** From the point feature \( F_p^i \) discussed above, we predict our interaction fields that jointly model human, object and their contacts, similar to CHORE [80]. Specifically, we predict the unsigned distances to human and object surfaces using \( f^u : F_p^i \mapsto \mathbb{R}^{5}_{\geq 0} \). This allows fitting the SMPL-Mesh mesh and object template by minimizing the predicted distances at mesh vertices. We can also infer contacts, as the points having small distances to both human and object surfaces. For more robust SMPL fitting [4] and modelling which body part the object point is in contact with, we predict SMPL part correspondence using \( f^p : F_p^i \mapsto \{1, 2, ..., K\} \) where \( K \) is the number of SMPL parts. For more accurate object fitting, we additionally predict rotation with \( f^R : F_p^i \mapsto \mathbb{R}^{3 \times 3} \) and translation with \( f^t : F_p^i \mapsto \mathbb{R}^3 \). The predicted \( 3 \times 3 \) matrix is projected to \( SO(3) \) using symmetric orthogonalization [44]. At test time, we first use \( f^u \) to find points on the object surface [16] and take the average rotation and translation predictions of these points as the object pose. To handle occlusions, we also predict the object visibility field using \( f^vis : F_p^i \mapsto [0, 1] \). The visibility is useful to recover the object pose under occlusion, see more details in Sec. 3.4.

**Why use triplane to encode SMPL-T?** A direct alternative to triplane based SMPL-T Encoding is to find the closest point in SMPL-Mesh and concatenate that coordinate to the point features. But such a method is slow (computing point to surface distance) and does not allow flexible learning of local and global features. Another choice is to voxelize the SMPL-T mesh and extract point local features using IF-Nets [15] but such a method is still expensive. Therefore we chose the more efficient triplane representation to encode our estimated SMPL-Meshes.

**Implementation.** Our SMPL-Mesh feature extractor \( f^m(\cdot) \) is shared for three views and trained end to end with image encoder \( f^{enc}(\cdot) \) and other neural field predictors. At training time, we input the renderings from ground truth SMPL-Meshes and train the network to predict GT labels. At test time, we obtain the SMPL-Meshes using Eq. (1). In order to have smoother SMPL-Mesh feature in a sequence, we use SmoothNet [90] to smooth the optimized SMPL parameters. We evaluate this component and provide more implementation details in supplementary.

### 3.4. HVOP-Net: Human and Visibility aware Object Pose under occlusions

Our SIF-Net recovers translation and more accurate object pose. However, the object pose prediction under very heavy occlusions remains challenging because no image evidence from single frame is available for accurate prediction, see Fig. 5. Our key idea is to use the SMPL-Mesh and object pose from other visible frames to predict the object of occluded frames. To this end, we first predict object vis-
Object visibility score. Our visibility score denotes how much the object is visible in the input image. We train a visibility decoder \( f_{\text{vis}}(\cdot) \), a prediction head of SIF-Net, that takes a point feature \( \mathbf{F}_i^p \) as input and predicts visibility score \( v_i \in [0, 1] \) for frame \( i \). At test time, we first use the neural object distance predictor \( f^n \) to find object surface points [16] and then take the average visibility predictions of these points as the object visibility score for this image.

Object pose prediction under heavy occlusion. Our goal now is to predict accurate object pose for heavily occluded frames. We consider frames whose visibility score \( v_i \) is smaller than \( \delta = 0.5 \) as the occluded frames. Inspired by works from motion infill [23, 39] and synthesis [88, 93], we design our HVOP-Net that leverages transformer [72] and explicitly takes the human motion and object visibility into account to recover object pose under heavy occlusions.

More specifically, we first use a transformer \( f^s(\cdot) \) to aggregate temporal information of the SMPL-T poses: \( f^s : \mathbb{R}^{T \times |\Theta|} \rightarrow \mathbb{R}^{T \times D_h} \), where \( |\Theta| \) is the SMPL pose dimension and \( D_h \) is the hidden feature dimension. Similarly, we use a transformer \( f^o(\cdot) \) to aggregate temporal information of the object poses: \( f^o : \mathbb{R}^{T \times D_o} \rightarrow \mathbb{R}^{T \times D_h} \). Note here the SMPL-T transformer \( f^s \) attends to all frames while the object transformer \( f^o \) only attends to frames where object is visible \((v_i \geq \delta)\). We then concatenate the SMPL-T and object features and use another transformer \( f^{\text{comb}} \) to aggregate both human and object information and predict the object poses: \( f^{\text{comb}} : \mathbb{R}^{T \times (D_h + D_o)} \rightarrow \mathbb{R}^{T \times D_o} \). The joint transformer \( f^{\text{comb}} \) attends to all frames. Our experiments show that our HVOP-Net is important to accurately predict object pose under heavy occlusions, see Fig. 5.

Implementation. The visibility decoder \( f_{\text{vis}} \) is trained end to end with other SIF-Net components using L2 loss. The GT visibility score is computed as the number of visible object pixels (from GT object rendering). To train our HVOP-Net, we randomly zero out the object pose for a small clip of the input sequence and provide ground truth SMPL and object poses for other frames as input. We train our network to accept input sequence of a fixed length but at test time, the sequence can have various length and object occlusion can last for a longer time. To this end, we use an auto-regressive algorithm to recover the object pose of a full video, similar to [89]. At test time, we use SIF-Net object pose predictions and zero out highly occluded frames based on the predicted visibility scores. We empirically find that having smooth object pose as input is helpful for more accurate prediction. Hence we use SmoothNet [90] to smooth SIF-Net object pose predictions before inputting them to HVOP-Net. The evaluation of this component and more training details are described in our supplementary.

3.5. Visibility aware joint optimization

To obtain SMPL and object meshes that align with input images and satisfy contact constraints, we leverage our network predictions from Sec. 3.3 and Sec. 3.4 to formulate a robust joint optimization objective. Our goal is to obtain an optimal set of parameters \( \Phi = (\Theta, \beta, R^o, T^o) \) for SMPL pose, shape, object rotation and translation respectively. We initialize the SMPL parameters from our estimated SMPL-T (Eq. (1)) and object parameters from our HVOP-Net predictions (Sec. 3.4). Inspired by CHORE [80], our energy function consists of human data \( E^h_{\text{data}} \), object data \( E^o_{\text{data}} \), contact data \( E^c_{\text{data}} \) and SMPL pose prior term \( E_{\text{reg}} \):

\[
E_{\Phi} = E^h_{\text{data}} + E^o_{\text{data}} + E^c_{\text{data}} + E_{\text{reg}}. \tag{2}
\]

here \( E_{\text{reg}} \) is a body pose and shape prior loss [49]. We explain other loss data terms next.

Human data term. \( E^h_{\text{data}} \) minimizes the discrepancy between SIF-Net prediction and SMPL meshes as well as a temporal smoothness error: \( E^h_{\text{data}}(\Theta, \beta) = \sum_{i=1}^T L^h_{\text{neural}}(\theta_i, \beta) + \lambda_{\text{occl}} L_{\text{accel}}(\Theta) \), where \( L_{\text{accel}} \) is the same used in Eq. (1). \( f^h \) pushes the SMPL vertices to the zero-level set of the human distance field represented by neural predictor \( f^h_{\text{vis}} \) and forces correct SMPL part locations predicted by \( f^p_{\text{vis}} \):

\[
E^h_{\text{neural}}(\Theta, \beta) = \sum_{i=1}^T \left( \sum_{p \in H(\Theta, \beta)} (\lambda_h \min(f^h_{\text{vis}}(\mathbf{F}_i^p), \delta_h) + \lambda_p L_p(l_p, f^p(\mathbf{F}_i^p))) \right) \tag{3}
\]

here \( l_p \) is the predefined SMPL part label [4] of SMPL vertex \( p \) and \( L_p \) is the categorical cross entropy loss function. \( \delta_h \) is a small clamping value.

Object data term. We transform the object template vertices \( O \in \mathbb{R}^{3 \times N} \) using pose parameters of frame \( i \) by: \( O'_i = R^o_i O + t^o_i \). Intuitively, the object vertices should lie on the zero-level set of the object distance field represented by \( f^o_{\text{vis}} \) and the rendered silhouette should match the 2D object mask \( M^o_i \). Hence we formulate the loss as:

\[
E^o_{\text{data}}(R^o, T^o) = \sum_{i=1}^T v_i \left( \sum_{p \in O'_i} \lambda_o \min(f^o_{\text{vis}}(\mathbf{F}_i^p), \delta_o) + \lambda_{\text{occl}} L_{\text{occl-sil}}(O'_i, M^o_i) + \lambda_{\text{smooth}} L_{\text{smooth}} \right) \tag{4}
\]

where \( v_i \) is the predicted object visibility score described in Sec. 3.4. This down-weights the loss values of network predictions for frames where object is occluded and allows more temporal regularization. \( L_{\text{occl-sil}} \) is an occlusion-aware silhouette loss [91] and \( L_{\text{smooth}} \) is a temporal smoothness loss applied to object vertices \( O'_i \), similar to \( L_{\text{accel}} \) in Eq. (1).
Contact data term. The contact data term [80] minimizes the distance between human and object points that are predicted to be in contact:

\[
E^c_{\text{data}}(\mathcal{R}^n, T^n) = \lambda c \sum_{i=1}^{T} \sum_{j=1}^{K} d(H^c_j(\theta_i, \beta), O^c_{ij})
\]  

(5)

here \(d(\cdot, \cdot)\) is chamfer distance. We consider human points on the \(j^{th}\) body part of SMPL mesh \(H_i\) of frame \(i\) (denoted as \(H_{ij}\)) are in contact when their predicted distance to the object is smaller than a threshold: \(H^c_j(\theta_i, \beta) = \{p|p \in H_i \text{ and } f^u_i(\mathbf{F}_i^p) \leq \epsilon \}\). Similarly, we find contact points on object meshes with \(O^c_{ij} = \{p|p \in O_i \text{ and } f^u_i(\mathbf{F}_i^p) \leq \epsilon \text{ and } f^u_i(\mathbf{F}_i^p) = j\}\).

Please see Supp. for more details about loss weights \(\lambda_c\).

4. Experiments

In this section, we first compare our method against existing approaches on tracking human and object and then evaluate the key components of our methods. Our experiments show that our method clearly outperforms existing joint human object reconstruction method and our novel SMPL-T conditioned interaction fields (SIF-Net) as well as human and visibility aware object pose prediction (HVOP-Net) works better than existing state of the art methods.

Baselines. (1) Joint human and object tracking. We compare against PHOSA [91] and CHORE [80] in the joint human-object reconstruction task. (2) Object pose prediction. Our HVOP-Net leverages nearby (un-occluded) frames to predict the object pose of occluded frames. We compare this with a simple baseline that linearly interpolates the object pose between visible frames to recover occluded poses. We also find similarity between our task and motion smoothing/infilling. Hence we compare our HVOP-Net with SoTA smoothing [90] and infill method [39].

Datasets. We conduct experiments on the BEHAVE [6] and InterCap [35] dataset. (1) BEHAVE [6] captures 7 subjects interacting with 20 different objects in natural environments and contains SMPL and object registrations annotated at 1fps. We use the extended BEHAVE dataset, which registers SMPL and object for BEHAVE sequences at 30 fps. We follow the official split [80] with 217 sequences for training and 82 for testing. (2) InterCap [35] is a similar dataset that captures 10 subjects interacting with 10 different objects. The dataset comes with pseudo ground truth SMPL and object registrations at 30fps. We train our model on sequences from subject 01-08 (173 sequences) and test on sequences from subject 09-10 (38 sequences).

We compare with CHORE [80] on the full test set of both datasets. PHOSA [91] optimizes hundreds of random object pose initialization per image hence the inference speed is very slow (2min/image), which makes it infeasible to run on full video sequences. Hence we compare with PHOSA on key frames only, denoted as BEHAVE\(^*\) (3.9k images) and InterCap\(^*\) (1.1k images) respectively. Due to the large number of frames in the full BEHAVE test set (127k frames), we conduct other ablation experiments in a sub test set (42k frames) of BEHAVE.

Evaluation metrics. (1) Joint human-object tracking. We evaluate the performance of SMPL and object reconstruction using Chamfer distance between predicted SMPL and object meshes, and the ground truth. CHORE [80] uses Procrustes alignment on combined SMPL and object meshes for each frame before computing errors. However, this does not reflect the real accuracy in terms of the relative translation between nearby frames in a video. Inspired by the world space errors proposed by SPEC [42] and GLAMR [89], we propose to perform joint Procrustes alignment in a sliding window, as also used in SLAM evaluations [31, 66]. More specifically, we combine all SMPL and object vertices within a sliding window and compute a single optimal Procrustes alignment to the ground truth vertices. This alignment is then applied to all SMPL and object vertices within this window and Chamfer distance of SMPL and object meshes are computed respectively. We report both the errors using per-frame alignment (w=1) and alignment with a sliding window of 10s (w=10).

(2) Object only evaluation. For experiments evaluating object pose only, we evaluate the rotation accuracy using rotation angle [32]. The object translation error is computed as the distance between reconstructed and GT translation. We report all errors in centimetre in our experiments.

4.1. Evaluation of tracking results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Methods</th>
<th>Align w=1 SMPL ↓</th>
<th>Obj ↓</th>
<th>Align w=10 SMPL ↓</th>
<th>Obj ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEHAVE</td>
<td>CHORE</td>
<td>5.55</td>
<td>10.02</td>
<td>18.33</td>
<td>20.32</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>5.25</td>
<td>8.04</td>
<td>7.81</td>
<td>8.49</td>
</tr>
<tr>
<td>BEHAVE*</td>
<td>PHOSA</td>
<td>12.86</td>
<td>26.90</td>
<td>27.01</td>
<td>59.08</td>
</tr>
<tr>
<td></td>
<td>CHORE</td>
<td>5.54</td>
<td>10.12</td>
<td>21.28</td>
<td>22.39</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>5.24</td>
<td>7.89</td>
<td>8.24</td>
<td>8.49</td>
</tr>
<tr>
<td>InterCap</td>
<td>CHORE</td>
<td>7.12</td>
<td>12.59</td>
<td>16.11</td>
<td>21.05</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>6.76</td>
<td>10.32</td>
<td>9.35</td>
<td>11.38</td>
</tr>
<tr>
<td>InterCap*</td>
<td>PHOSA</td>
<td>11.20</td>
<td>20.57</td>
<td>24.16</td>
<td>43.06</td>
</tr>
<tr>
<td></td>
<td>CHORE</td>
<td>7.01</td>
<td>12.81</td>
<td>16.10</td>
<td>21.08</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>6.78</td>
<td>10.34</td>
<td>9.35</td>
<td>11.54</td>
</tr>
</tbody>
</table>

Table 1. Human and object tracking results on BEHAVE [6] and InterCap [35] datasets (unit: cm). \(^*\) denotes key frames only. w is the temporal window size used for Procrustes alignment where w=1 means per-frame Procrustes and w=10 means alignment over a sliding window of 10s. We can see our method clearly outperforms baseline PHOSA [91] and CHORE [80] in all metrics.

We compare our human and object tracking results against baseline PHOSA [91] and CHORE [80] and report the errors in Tab. 1. Note that the comparison with PHOSA
Figure 3. Comparison with PHOSA [91] and CHORE [80] on BEHAVE [6] (row 1-2) and InterCap [35] (row 3-4). PHOSA’s object pose optimization often gets stuck in local minima due to heavy occlusions. CHORE also fails to predict accurate object pose as it does not fully explore the human, temporal and visibility information while our method can robustly track human and object in these challenging cases.

and CHORE is not strictly fair as they do not use temporal information. Nevertheless they are our closest baselines and we show that our method outperforms them using per-frame alignment and is significantly better under a more relevant evaluation metric (align w=10) for video tracking. We also show qualitative comparisons in Fig. 3. It can be seen that both PHOSA and CHORE are not able to accurately capture the object under heavy occlusions while our method is more robust under these challenging occlusion cases.

4.2. Importance of SMPL-T conditioning

We propose to condition our interaction field predictions on SMPL-T meshes, we evaluate this for the joint tracking task in Tab. 2. Our SMPL-T conditioning allows us to obtain consistent relative translation instead of predicting the human at fixed depth. This significantly reduces the error when evaluating using alignment of temporal sliding win-

<table>
<thead>
<tr>
<th>Method</th>
<th>w/o SMPL-T</th>
<th>w/o HVOP-Net</th>
<th>w/o joint opt.</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMPL ↓</td>
<td>14.40</td>
<td>8.05</td>
<td>8.20</td>
<td>8.03</td>
</tr>
<tr>
<td>Obj. ↓</td>
<td>17.29</td>
<td>9.36</td>
<td>16.02</td>
<td>8.23</td>
</tr>
</tbody>
</table>

Table 2. Ablation studies. We report the joint tracking error (cm) after an alignment window of 10s. It can be seen that our proposed SMPL-T conditioning, HVOP-Net and joint optimization are important to achieve the best results.

dow, see Tab. 2 and qualitative examples in Supp.

Note that the SIF-Net prediction relies on the estimated SMPL-T discussed in Sec. 3.2. We also evaluate how the noisy SMPL-T predictions can affect the joint tracking performance. We input GT SMPL to our SIF-Net and HVOP-Net to predict the object pose and perform joint optimization. The object errors (Chamfer distance in cm) are: 17.29 (w/o SMPL-T cond.), 8.23 (w/ SMPL-T cond., ours), 6.50 (GT SMPL). Our method is robust and is close to the ideal lower bound with GT SMPL.
4.3. Importance of HVOP-Net

We propose a human and visibility aware object pose prediction network (HVOP-Net) to reason about the object under heavy occlusions. Without this component, the object error is much higher (Tab. 2 column 3), which suggests the importance of our HVOP-Net.

For the object pose prediction task, there are other similar alternatives to HVOP-Net: 1). Linear interpolation (SLERP), infill the object pose of invisible frames using spherical linear interpolation between two visible frames. 2). SmoothNet [90], a fully connected neural network trained to smooth the object motion. 3). CMIB [39], a SoTA method for human motion infilling. To evaluate the effectiveness of our HVOP-Net, we replace HVOP-Net with these methods and run the full tracking pipeline respectively. The SMPL errors are similar (deviate < 0.1cm) as HVOP-Net only affects objects. We separate object error into rotation (angle distance) and translation, and further analyse the error under varying occlusion for CHORE, our method (SMPL-T + HVOP-Net) and its variants (SMPL-T + SmoothNet/SLERP/CMIB) in Fig. 4.

All methods except CHORE predict similar translation (due to SMPL-T cond.) but our HVOP-Net obtains clearly more robust rotation under heavy occlusions. SmoothNet smooths the object motion but cannot correct errors from long-term occlusions. SLERP and CMIB [39] are able to correct some pose errors but do not take the human context into account hence cannot handle heavy occlusions very well. Our method leverages the human motion and object pose from visible frames hence achieves the best result. We show one example where our HVOP-Net corrects the erroneous raw prediction in Fig. 5. Please see our supplementary for more examples.

4.4. Generalization

To verify the generalization ability of our method, we apply our model trained on BEHAVE to the NTU-RGBD [47] dataset. We leverage Detectron [79], interactive [64] and video [14] segmentation to obtain the input human and object masks. Two example comparisons with CHORE [80] are shown in Fig. 6. We can see our method generalizes well to NTU-RGBD and works better than CHORE. Please see Supp. for evaluation details and more comparisons.

5. Limitations

Although our method works robustly under heavy occlusions, there are still some limitations. For instance, we assume known object templates for tracking. An interesting direction is to build such a template from videos as demonstrated by recent works [78,82–84]. Our method may fail under challenging cases such as significant object pose change under heavy occlusions. It can also make noisy pose predictions when the object is symmetric or the pose is uncommon. Example failure cases are shown in Supp.

6. Conclusion

We present a novel method for tracking human, object and realistic contacts between them from monocular RGB cameras. Our first contribution is a SMPL-T conditioned neural field network that allows consistent and more accurate 3D reconstruction in a video sequence. Our second contribution is a human motion and object visibility aware network that can recover the object pose under heavy occlusions. Our experiments show that our method significantly outperforms state of the art methods in two datasets. Our extensive ablations show that our SMPL-T conditioning and HVOP-Net are important for accurate tracking. We also show that our method generalizes well to another dataset it is not trained on. Our code and model are released to promote future research in this direction.

Acknowledgements. We thank RVH group members [1] for their helpful discussions. We also thank reviewers for their feedback which improves the manuscript. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans), and German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. Gerard Pons-Moll is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.
Appendices

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 [80] on NTU-RGBD [47] dataset. We end with discussions of failure cases and future works.

Appendix A. Implementation details

A.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: \( \lambda_{\text{2D}} = 0.09, \lambda_{\text{reg}} = 1.0 \times 10^{-5}, \lambda_u = 25, \lambda_\text{pt} = 900 \). We optimize the parameters until convergence with a maximum iteration of 1000.

A.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. 7. We discuss our network architecture and training details next.

Network architecture. We use the stacked hourglass network [51] for both RGB image encoder \( f^{\text{enc}} \) and SMPL-rendering encoder \( f^{\text{tri}} \). We use 3 stacks for \( f^{\text{tri}} \) and the output feature dimension is \( d_{\text{tri}} = 64 \). 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 \( f^{\text{enc}} \) but the feature dimension is \( d^{\text{enc}} = 256 \). 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: \( d = (d_f^{\text{tri}} + d_{\text{tri}}^{\text{tri}}) \times 3 + d_{\text{enc}}^{\text{enc}} + d_{\text{enc}}^{\text{enc}} + 3 = 611 \), here \( d_{\text{tri}} = 32, d_{\text{enc}} = 64 \). 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.

Training. All feature encoders and decoders are trained end to end with the loss: \( L = \lambda_u (L_u + L_v) + \lambda_p L_p + \lambda_R L_R + \lambda_c L_c + \lambda_v L_v \). Here \( L_{ui} \) is the L1 distance between ground truth and predicted unsigned distance to human or object surface [80]. \( L_p \) 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: \( \lambda_u = 1.0, \lambda_R = 0.006, \lambda_c = 500, \lambda_v = \lambda_o = 1000 \).

Figure 7. 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 \( F^A, F^B, F^C \) 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.

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.

A.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 [96] to represent the rational 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 (MHSAta) module [72] 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 MHSAta module with 2 heads. The joint transformer \( f^{\text{comb}} \) consists of 4 layers of MHSAta module with 1 head only. GeLU activation is used in all MHSAta 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).

A.4. SmoothNet for SMPL-T and object

We use SmoothNet [90] 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 [90], we train both models on the predictions from the BEHAVE [6] training set. Note that we do not fine-tune them on InterCap [35] dataset. We evaluate this component in Sec. B.3.

A.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: $\lambda_{\text{reg}} = 2 \times 10^{-4}, \lambda_{\text{ah}} = 10^4, \lambda_{o} = 10^4, \lambda_{c} = 225, \lambda_{\text{occ}} = 9 \times 10^{-4}, \lambda_{\text{ao}} = 900$, where $\lambda_{c}$ is the loss weight for the contact data term defined in Eq. 5.

Appendix B. Additional ablation results

B.1. Further evaluation of SMPL-T conditioning

We show some example images from one sequence in Fig. 8 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.

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. 3. We can clearly see that our SMPL feature improves both the raw object pose prediction and distance fields (results after optimization are also improved).

<table>
<thead>
<tr>
<th>Method</th>
<th>Raw prediction</th>
<th>After opt. w=10</th>
<th>Chamfer</th>
<th>v2v</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o SMPL-T</td>
<td>CD↓ v2v↓ transl↓</td>
<td>SMPL↓ obj↓</td>
<td>5.56</td>
<td>14.28</td>
<td>14.40</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
<td>3.98</td>
<td>12.34</td>
<td>9.53</td>
</tr>
</tbody>
</table>

Table 3. 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).

B.2. Comparing different pose prediction methods

We show some example comparisons of different object pose prediction methods under heavy occlusions in Fig. 10.

<table>
<thead>
<tr>
<th>Method</th>
<th>Chamfer</th>
<th>v2v</th>
<th>Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o SmoothNet</td>
<td>8.71</td>
<td>9.84</td>
<td>1.38</td>
</tr>
<tr>
<td>w/ SmoothNet</td>
<td><strong>8.01</strong></td>
<td><strong>9.12</strong></td>
<td><strong>1.18</strong></td>
</tr>
</tbody>
</table>

Table 4. Ablate SMPL SmoothNet (errors in cm). We can see that SmoothNet [90] improves the overall smoothness and slightly reduces the pose errors.

We compare our method against: 1). Raw prediction from our SIF-Net. 2). Linearly interpolate the occluded poses from visible frames (SLERP). 3). CMIB [39], 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. B.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.

B.3. Evaluating SmoothNet

SmoothNet [90] 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. 4. We can see that SmoothNet reduces the SMPL error slightly.
Table 5. Ablate SmoothNet for object pose prediction (errors in cm). We can see our pose prediction (c) is better than SmoothNet [90] (b). Combing both we obtain the best result (d).

<table>
<thead>
<tr>
<th>Method</th>
<th>Chamfer</th>
<th>v2v</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Raw prediction</td>
<td>5.03</td>
<td>10.39</td>
<td>10.01</td>
</tr>
<tr>
<td>b. Raw + SmoothNet</td>
<td>4.22</td>
<td>8.60</td>
<td>10.16</td>
</tr>
<tr>
<td>c. Raw + our pose pred.</td>
<td>4.09</td>
<td>8.02</td>
<td>10.20</td>
</tr>
<tr>
<td>d. Our full model</td>
<td>3.62</td>
<td>7.20</td>
<td>9.96</td>
</tr>
</tbody>
</table>

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. 5. It can be seen that our method (Tab. 5c) works better than SmoothNet (Tab. 5b) on raw predictions. Nevertheless, with smoothed pose after SmoothNet, our method achieves the best result (Tab. 5d).

B.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 (∼12s/frame) [80], 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.

Appendix C. 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 [79] to obtain the human masks. We manually segment the object in the first frame using interactive segmentation [64] (<1min/image) and then use video segmentation [14] to propagate the masks. The overhead of 1min/video manual label is small.

We show more results from our method on NTU-RGBD dataset [47] and compare against CHORE [80] in Fig. 11. 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.

![Figure 9. 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.](image)

Appendix D. 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 [78,82–84]. 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 [46] or object articulation [81] 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. 9. 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. 9 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. 9 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 correct (Fig. 9 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 [94] can be helpful.

References

[25] Zhaoxin Fan, Yazhi Zhu, Yulin He, Qi Sun, Hongyan Liu, and Jun He. Deep Learning on Monocular Object Pose


[59] Chris Rockwell and David F. Fouhey. Full-body awareness from partial observations. In *ECCV*, 2020. 2


Figure 10. Comparing different object pose prediction methods under heavy occlusions. Raw prediction is from our SIF-Net output, SLERP denotes linear interpolation and CMIB is from [39]. 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 11. Comparing our method with CHORE [80] on NTU-RGBD [47] 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.