BEHAVE: Dataset and Method for Tracking Human Object Interactions

Bharat Lal Bhatnagar1,2, Xianghui Xie2, Ilya A. Petrov1, Cristian Sminchisescu3, Christian Theobalt2, and Gerard Pons-Moll1,2

1University of Tübingen, Germany
2Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
3Google Research

{i.petrov, gerard.pons-moll}@uni-tuebingen.de, {bbhatnag, xxie, theobalt}@mpi-inf.mpg.de, sminchisescu@google.com

Abstract

Modelling interactions between humans and objects in natural environments is central to many applications including gaming, virtual and mixed reality, as well as human behavior analysis and human-robot collaboration. This challenging operation scenario requires generalization to vast number of objects, scenes, and human actions. Unfortunately, there exist no such dataset. Moreover, this data needs to be acquired in diverse natural environments, which rules out 4D scanners and marker based capture systems. We present BEHAVE dataset, the first full body human-object interaction dataset with multi-view RGBD frames and corresponding 3D SMPL and object fits along with the annotated contacts between them. We record ~15k frames at 5 locations with 8 subjects performing a wide range of interactions with 20 common objects. We use this data to learn a model that can jointly track humans and objects in natural environments with an easy-to-use portable multi-camera setup. Our key insight is to predict correspondences from the human and the object to a statistical body model to obtain human-object contacts during interactions. Our approach can record and track not just the humans and objects but also their interactions, modeled as surface contacts, in 3D. Our code and data can be found at: http://virtualhumans.mpi-inf.mpg.de/behave.

1. Introduction

The last decade has seen rapid progress in modelling the appearance of humans ranging from body pose, shape [52, 58, 60, 61, 81], faces [74] and even detailed clothing [5, 7, 11, 57, 65]. With various practical use cases like virtual try-on, personalised avatar creation, and several applications in augmented and mixed reality, or human-robot collaboration, the focus on humans is justified. Beyond modelling appearance, few methods have focused on capturing and synthesizing human interactions (human-object/scene interaction). There exists work to capture humans in a static 3D scene [33], even without using external cameras [30], and work to synthesize static poses [34, 49], or full body movement [32, 30, 68] in a 3D scene.

These methods show growing interest in modelling human behavior, highlighting a need to capture real human interactions. Existing methods [32, 68] however are learned from high quality curated data captured using optical marker based motion capture systems or wearable sensors. Unfortunately, such commercial systems are expensive, drastically limit the interactions that can be captured, and often fail when tracking humans and objects under occlusion. In addition, the recording volume is spatially confined and difficult to re-locate, thus limiting the activities, scenes, and objects that can be captured. Wearable sensors [30] are not restricted in volume, but close range interaction can not be accurately captured. Altogether, the
lack of diverse 3D interaction data, and the lack of accurate and flexible capture methods both constitute barriers in modelling human behavior.

With the goal of simplifying the data capture process and hence allowing faster progress in the field, we propose BEHAVE, a method to capture diverse 3D human interactions in natural environments, using a setup comprising of portable, cheap, and easy to use RGBD cameras. Tracking human interactions from sparse consumer grade cameras is however extremely challenging. Depth data is inherently noisy and incomplete. Moreover, the person and object occlude each other frequently during interactions. Furthermore, capturing interactions requires estimating human-object contacts accurately, which is difficult because contacts represent small regions in the image, close to the observable (resolution) limit. This requires innovation that goes significantly beyond the current state of the art trackers. We propose to track the human using a parametric human model (such as SMPL [52]) and track objects using template meshes. Naively fitting the human model and an object 3D template to the point-cloud completely fails due to the aforementioned challenges. Our key idea is to train a neural model which jointly completes the human and object shape, represented with implicit surfaces, while predicting a correspondence field to the human, as well as an object orientation field. These rich outputs allow us to formulate a powerful human-object fitting objective which is robust to missing data, noise and occlusion.

To train and evaluate BEHAVE, we capture the largest dataset of human-object interactions in natural environments. The BEHAVE dataset contains 20 3D objects, 8 subjects (5 male, 3 female), 5 different locations and totals around 15.2k frames of recording. We provide ground truth SMPL and 3D object meshes as well as contacts. Our contributions can be summarized as follows:

- We propose the first approach that can accurately 3D track humans, objects and contacts in natural environments using multi-view RGBD images.
- We collect the largest dataset of multi-view RGBD sequences and corresponding human models, object and contact annotations. See Sec. 3 for details regarding its usefulness to the community.
- Since there exists no publicly available code and datasets to accurately track human-object interactions in natural environments, we will release our code and data for further research in this direction.

2. Related Work

In this section, we first briefly review work focused on object and human reconstruction, in isolation from their environmental context. Such methods focus on modelling appearance and do not consider interactions. Next, we cover methods focused on humans in static scenes and finally discuss closer-related work to ours, for modelling dynamic human-object interactions.

2.1. Appearance modelling: Humans and objects without scene context

**Human reconstruction and performance capture** Perceiving humans from monocular RGB data [12, 29, 31, 41, 43, 44, 58, 59, 64, 87] and under multiple views [37–40, 62] settings has been widely explored. Recent work tends to focus on reconstructing fine details like hand gestures and facial expressions [20, 25, 85, 91], self-contacts [27, 54], interactions between humans [26], and even clothing [6, 11]. These methods benefit from representing human with parametric body models [52, 58, 81], thus motivating our use of recent implicit diffused representations [8, 10] as backbone for our tracker.

Following the success of pixel-aligned implicit function learning [64, 65], recent methods can capture human performance from sparse [38, 80] or even a single RGB camera [47, 48]. However, capturing 3D humans from RGB data involves a fundamental ambiguity between depth and scale. Therefore, recent methods use RGBD [56, 69, 73, 76, 84] or volumetric data [9, 10, 19] for reliable human capture. These insights motivate us to build novel trackers based on multi-view RGBD data.

**Object reconstruction** Most existing work on reconstructing 3D objects from RGB [21, 46, 53, 75, 78] and RGBD [45, 55, 82] data does so in isolation, without the human involvement or the interaction. While challenging, it is arguably more interesting to reconstruct objects in a dynamic setting under severe occlusions from the human.

2.2. Interaction modelling: Humans and objects with scene context

**Humans in static scenes** Modelling how humans act in a scene is both important and challenging. Tasks like placement of humans into static scenes [34, 49, 90], motion prediction [15, 32] or human pose reconstruction [16, 33, 77, 86, 89] under scene constrains, or learning priors for human-object interactions [66], have been investigated extensively in recent years. These methods are relevant but restricted to modelling humans interacting with static objects. We address a more challenging problem of jointly tracking human-object interactions in dynamic environments where objects are manipulated.

**Dynamic human object interactions** Recently, there has been a strong push on modeling hand-object interactions based on 3D [42, 72], 2.5D [13, 14] and 2D [22, 24, 28, 35, 83]...
### 3. BEHAVE Dataset

We present BEHAVE dataset, the largest dataset of human-object interactions in natural environments, with 3D human, object and contact annotation, to date. See Tab. 1 for comparison with other datasets. Our dataset contains multi-view RGBD frames, with accurate pseudo-ground truth SMPL [52], object fits, human and object segmentation masks, and contact annotations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NTU [51]</td>
<td>✓</td>
<td>Jts.</td>
<td>X</td>
<td>NA</td>
<td>***</td>
</tr>
<tr>
<td>PiGr [66]</td>
<td>✓</td>
<td>Jts.</td>
<td>X</td>
<td>NA</td>
<td>**</td>
</tr>
<tr>
<td>GRAB [72]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>***</td>
<td>*</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>**</td>
<td>***</td>
</tr>
</tbody>
</table>

Table 1. We compare the proposed BEHAVE dataset with existing ones containing human-object interactions. Our criteria are based on availability of RGB input, 3D human, 3D contact with the object, quality (more stars, better), and scalability to capture at diverse locations (more stars, better). NTU-RGBD [51] and PiGraphs [66] do not provide full 3D human and object contacts and are hence unsuitable for modelling dynamic 3D interactions. GRAB [72] uses a marker based capture system and hence contains the highest quality data but this also makes it difficult to scale. PROX [33] is easier to scale as it uses a single Kinect based capture setup (although, scene needs to be pre-scanned) but this reduces the overall quality. More importantly it does not contain dynamic interactions. Ours is the first dataset that captures dynamic human-object interactions in diverse environments.

Data. Although powerful, these methods are currently restricted to modelling only hand-object interactions. In contrast, we are interested in full body capture. Methods for dynamic full body human object interaction approach the problem via 2D action recognition [36, 51] or reconstruct 3D object trajectories during interactions [23]. Despite being impressive, such methods either lack full 3D reasoning [36, 51] or are limited to specific objects [23].

More recent work reconstructs and tracks human-object interactions from RGB [71] or RGBD streams [70], but does not consider contact prediction, thus missing a component necessary for accurate interaction estimates.

Very relevant to our work, PHOSA [88] reconstructs humans and objects from a single image. PHOSA uses hand crafted heuristics, instance specific optimization for fitting, and pre-defined contact regions, which limits generalization to diverse human-object interactions. Our method on the other hand learns to predict the necessary information from data, making our models more scale-able. As shown in the experiments, the accuracy of our method is significantly higher to PHOSA.

**Recording multi-view RGBD data** We setup and calibrate 4 Kinects at 4 corners of our square recording volume where all interactions are performed by 8 subjects (5 male, 3 female). Interactions are captured at 5 disparate indoor locations with 20 commonly used, yet diverse objects: 5 different boxes, 2 chairs, 2 tables, crate, backpack, trash-can, monitor, keyboard, suitcase, basketball, exercise ball, yoga mat, stool and a toolbox. We include common interactions such as lifting, carrying, sitting, pushing and pulling with hands and feet, as well as free interactions. See our supplementary video for sample sequences. In total, our dataset contains 10.7k frames for training and 4.5k frames for testing respectively.

**Human segmentation and SMPL fitting** We segment the human in our images by running DetectronV2 [79] followed by manual correction with [67] on the segmentation masks. These masks are then used to segment multi-view depth maps and lift human point clouds from 2D to 3D. We use FrankMocap [63] to initialize SMPL’s pose from the images and then use instance specific optimization [6] to fit the SMPL model to the segmented human point cloud. For more accurate fitting, we additionally obtain the SMPL shape parameters of each subject from 3D scans using [9]. We report a chamfer error of 1.80cm between the segmented kinect point cloud and our SMPL fits.

**Object segmentation and fitting** To obtain object segmentation, we pre-scan objects using a 3D scanner [1, 3]. We then use multi-view object keypoints, marked manually by AMT [2] annotators in images, to optimize the 6D pose of the pre-scanned object mesh to the given frame. We obtain the chamfer error of 2.42cm between the segmented Kinect point cloud and object fit. The segmentation masks are then obtained by projecting fitted object meshes to the images.

**Contact annotation** Based on the pseudo-GT SMPL and object fits as described above, we automatically detect contacts if a point on the human surface (registered SMPL) is closer than 2cm to the object surface. For every object point, we store a binary contact label (whether there is a contact or not) and correspondence to the human (contact location on the surface).

See supplementary for more details on data acquisition.

**How will this dataset be useful to the community?** We devote significant effort in recording the largest, so far, dataset of natural, full body, day-to-day human interactions with common objects in different natural environments. We propose following challenges with BEHAVE dataset:
4. Method: Tracking human, object and contact

We present BEHAVE, a method to jointly track humans, objects and their interactions (represented as surface contacts) based on multi-view RGB-D input. We formulate our method as an extended per-frame registration problem: we register the human (using SMPL [52]) and the object (using its pre-scanned object mesh), and predict contacts as correspondences between SMPL and object meshes. See Fig. 3 for the overview of our method.

Our formulation must obey three properties, (i) the SMPL model $M(\cdot)$ should fit the human in the multi-view input, (ii) the object mesh $W^o$ should fit the input object and, (iii) SMPL model and object should satisfy contacts.

To facilitate joint reasoning of the human, object and contacts directly in 3D, we lift the human $S^h$, and object $S^o$ point clouds to 3D using multi-view depth and semantic segmentation. Our joint formulation fits SMPL $M(\cdot)$ and the object $W^o$ to multi-view RGB-D data at each time step, using explicit contacts. This takes the following form

$$E(\theta, \beta, R^o, t^o) = E^{\text{SMPL}}(S^h, M(\theta, \beta)) + E^{\text{obj}}(S^o, W^o) + E^{\text{contact}}(1^cW^o, M(\theta, \beta)).$$

The SMPL model is parameterized by pose $\theta$, and shape $\beta$. For notation brevity, we include the global SMPL translation into the pose parameters. We assume the template object $W$ be rigid and only estimate the rotation $R^o$, and translation $t^o$, to fit the object mesh $W^o = R^oW + t^o$, to the object point cloud.

The indicator matrix, $1^c$, selects the vertices on the object mesh $W^o$, that are in contact with the SMPL model. This ensures that contact locations on the object and the human mesh adequately align in 3D.

The term $E^{\text{SMPL}}(S^h, M(\theta, \beta))$ is designed to accurately fit SMPL to the human point cloud $S^h$. The term $E^{\text{obj}}(S^o, W^o)$ is designed to fit the object mesh to the object point cloud and $E^{\text{contact}}(1^cW^o, M(\theta, \beta))$ ensures that contacts between the human and object match (align). We explain each term in detail next.

4.1. Fitting human model to the human point cloud

Fitting SMPL to the human point cloud $S^h$ requires, (i) that distance between the SMPL model and the human point cloud should be minimized and (ii) the correct SMPL parts fit the corresponding body parts of the point cloud. The latter is important to avoid degenerate cases such as 180° flipped fitting, where the left hand is erroneously matched to the right side of the body or vice-versa [9]. With these considerations, we design our SMPL fitting objective as:

$$E^{\text{SMPL}} = d(S^h, M(\theta, \beta)) + E^{\text{corr}} + E^{\text{reg}},$$

where $d(S^h, M(\theta, \beta))$, $E^{\text{corr}}$, and $E^{\text{reg}}$ correspond to the distances of the fitted SMPL model from the human point cloud, error in the multi-view RGB-D data, and shape regularization, respectively.
where \( d(S^h, M(\theta, \beta)) \) minimizes the point-to-mesh distance between the input human point cloud \( S^h \) and the SMPL model. To avoid sub-optimal local minima during fitting [9, 10], we train a neural network that predicts dense correspondences from the input to the SMPL model. This ensures that correct SMPL parts explain corresponding input regions, using the term \( E^{\text{corr}} \).

Specifically, we train an encoder network similar to [17, 18] that takes the segmented and voxelized human \( S^h \) and object \( S^o \) point cloud as inputs, and generates a voxel aligned grid of features \( F = f_\phi^{\text{enc}}(S^h, S^o) \). We then sample \( N \) 3D query points, \( \{p_1, \ldots, p_N\}, p_i \in \mathbb{R}^3 \) and for each point \( p_i = (x, y, z) \) obtain the corresponding point feature \( F_i = F(x, y, z) \). We pass this point feature through a decoder network \( f_\phi^{\text{aff}} \) to predict the signed distance to object and human surfaces, \( v_i^o, u_i^h = f_\phi^{\text{aff}}(F_i), v_i^o, u_i^h \in \mathbb{R}, \) respectively. We use a second decoder network \( f_\phi^{\text{corr}} \) to predict the correspondence of point \( p_i \) to the SMPL model, \( c_i = f_\phi^{\text{corr}}(F_i), c_i \in \mathbb{R}^3 \).

\( E^{\text{corr}} \) enforces that the distance between the input point \( p_i \) and the corresponding point \( c_i \) after transforming it with the SMPL model is same as the distance predicted by the network \( u_i^h \). Under a slight abuse of notation we use \( M(c_i, \theta, \beta) \) to transform \( c_i \) with the SMPL function.

\[
E^{\text{corr}} = \sum_{i=1}^N ||p_i - M(c_i, \theta, \beta)||_2 - u_i^h, \quad (3)
\]

If the correspondences predicted by the network \( c_i \) deviate from the SMPL surface, these cannot be skinned using the SMPL model as its function is only defined on the body surface. To alleviate this issue, we use the LoopReg [10] formulation that allows us to pose and shape off-the-surface correspondences as well.

The final term \( E^{\text{reg}} = E^{\text{2D}} + E^\theta + E^\beta \), adds regularisation for SMPL joints, \( E^{\text{2D}} = \sum_{k=1}^K |\pi_k M^j(\theta, \beta) - J^k_{2\text{D}}|_2 \), where \( \pi_k \) is the camera projection matrix of camera \( k \), \( M^j(\cdot) \) are the 3D body joints and \( J^k_{2\text{D}} \) are the 2D joints detected in the \( k^{\text{th}} \) Kinect image. \( E^\theta \) and \( E^\beta \) are regularisation terms on SMPL pose and shape similar to [12].

### 4.2. Fitting the object mesh to the object point cloud

In order to fit the object mesh, we must ensure that distance from the input object point cloud to the object mesh is minimized. Minimizing this one-sided distance is necessary but not sufficient. Since severe occlusions are common in our interaction setting, large parts of object might be missing from the object point cloud, making fitting difficult. To alleviate this issue we must also ensure that all the vertices of the object mesh are correctly placed w.r.t. the input, even when the point cloud is incomplete. To do so, we take the object mesh vertices \( v_j^o \in W^o, j \in \{1, \ldots, L\} \) and obtain the corresponding point feature \( F_j \), same as Sec. 4.1, where \( L \) is the number of object mesh vertices. We then obtain the unsigned distances to the object and human surfaces using the point feature \( u_j^o, u_j^h = f_\phi^{\text{aff}}(F_j) \). Since \( v_j^o \) is a vertex on the object mesh, its distance to the object surface \( u_j^o \) must be zero for a correct fit. This allows us to accurately fit the object vertices to the point data even when corresponding parts are missing from the object point cloud.

\[
E^{\text{obj}} = d(S^o, W^o) + \sum_{j=1}^L |u_j^o|, \quad (4)
\]

where \( d(S^o, W^o) \) minimizes the point-to-mesh distance between the object point cloud and the object mesh, and the term \( \sum_{j=1}^L |u_j^o| \) uses implicit unsigned distance prediction to reason about missing object parts.
Predicting object orientation. Although the terms in Eq. (4) minimize the bi-directional distance between the object point cloud and the object mesh, they do not guarantee that parts of the object point cloud are explained by the semantically corresponding parts on the object mesh, e.g. in Fig. 4, the legs of the table are not aligned correctly. This issue can be fixed if we obtain the global object orientation during fitting. We represent the orientation of the object with the principal components obtained by running PCA on the object vertices.

We train a neural network \( f^o_\phi \), that uses the point feature \( F_j \) (same as Sec. 4.1) corresponding to each query point \( p_j \) and predicts the global orientation of the object \( a_j = f^o_\phi(F_j), a_j \in \mathbb{R}^3 \). We find that orientation prediction is unreliable if the query point is far from the object surface, hence we filter out points whose unsigned distance from the object surface \( u^o_j \), is greater than a threshold \( \epsilon = 2cm \). The global orientation of the object is obtained by averaging the orientation predictions from the filtered points, \( a^o = \frac{1}{M} \sum_{j=1}^{M} a_j \) where \( M \) is the number of filtered points. Next, we compute the relative rotation between the current object orientation \( a \) and the predicted object orientation \( a^o \), and use this to initialise the object rotation \( R^o = a^o(a^T \hat{a})^{-1} \hat{a}^T \). We further run SVD on \( R^o \) and only keep the rotation matrix.

Initialising \( R^o \) with the network predicted object orientation is crucial to avoid local minima during object fitting, as can be seen in Fig. 4 and Tab. 3.

### 4.3. Refining human & object models using contacts

Our formulation above gives reasonably good human and object fits but does not ensure that human and object meshes satisfy the contacts predicted by the network. This often leads to floating objects and hovering hands see Fig. 5, as human and object models are not in contact. In this section we explicitly optimize the human and object meshes to fit the contacts predicted by the network. We model contacts as vertices in the registered object mesh \( v^o_j \in W^o \),

that are very close to the input human \( u^h_j < \epsilon \) and object \( u^o_j < \epsilon \) surface. Similarly to Sec. 4.2, we use \( f^\alpha_\phi \) to obtain the unsigned distances \( u^o_j, u^h_j \) and \( f^\text{corr}_\phi \) to obtain the correspondences \( c_j \) of these points to the SMPL model, respectively. In order to filter query points close to human and object surfaces we compute a binary indicator matrix \( \mathbb{1} \in \mathbb{R}^N \) such that \( \mathbb{1}_j = 1 \text{ iff } u^h_j < \epsilon, u^o_j < \epsilon \).

\[
E^{\text{contact}} = \sum_{j=1}^{N} \mathbb{1}_j |v^o_j - M(c_j, \theta, \beta)|_2.
\]

\( E^{\text{contact}} \) allows us to jointly optimise the SMPL model and the object parameters \( R^o, t^o \) to satisfy the contacts predicted by the network.

### 4.4. Network training

In this section we elaborate on training our networks.

**Feature encoding.** We use a 3D CNN similar to IFNet [17] to obtain a voxel aligned multi-scale grid of features \( F = f^\text{enc}_\phi(S^h, S^o) \).

**Unsigned distance prediction.** To train the network \( f^\alpha_\phi \), we sample \( N \) query points \( \{p_1, \ldots, p_N\} \) in 3D. For each query point \( p_j \) we obtain its point feature \( F_j \) (Sec. 4.1) and use this to predict the unsigned distance [18] to human and object surface \( u^o_j, u^h_j = f^\alpha_\phi(F_j) \).

We jointly train \( f^\text{enc}_\phi, f^\alpha_\phi \) with standard L2 loss. The GT for \( u^o_j, u^h_j \) is easily available as our dataset contains GT SMPL and object fits allowing us to obtain GT distance of point \( p_j \) from the SMPL and object mesh.

**SMPL correspondence prediction.** To train \( f^\text{corr}_\phi \), we use the point feature \( F_j \) of sampled query point \( p_j \) to predict its correspondence to the SMPL model \( c_j = f^\text{corr}_\phi(F_j) \).

We jointly train \( f^\text{enc}_\phi, f^\alpha_\phi \) using a standard L2 loss. Since we have the GT SMPL fit in our dataset we simply find the closest SMPL surface point for the query point \( p_j \) and use this as the GT correspondence.
5. Experiments

In this section we compare our approach with existing methods. Our experiments show that we clearly outperform existing baselines. Next, we ablate our design choices and highlight the importance of contact and object orientation prediction in capturing human-object interactions.

5.1. Comparing with PHOSA

We find PHOSA [88], a method to reconstruct humans and objects from a single image, quite relevant to our work. Although PHOSA uses only a single image whereas we use multi-view images, thus giving our method an advantage, it is still the closest competing method. We run Procrustes alignment on PHOSA results to remove depth ambiguity. It should be noted that PHOSA depends on pre-defined fixed contact regions whereas our approach can freely predict full-body contacts and PHOSA uses hand crafted heuristics to model contacts whereas our approach learns contact modelling from data, making our method more scalable. We compare our method with PHOSA in Fig. 6 and Tab. 2, and clearly outperform it.

5.2. Why not fit human and object models directly to point clouds?

Since there are no existing methods that can jointly track humans, objects and the contacts from a multi-view input, we create an obvious baseline where we fit the SMPL and object meshes directly to the input point cloud. We show (Tab. 2) that direct fitting easily gets stuck in local minima.

<table>
<thead>
<tr>
<th>Method</th>
<th>SMPL v2v (cm)</th>
<th>obj. v2v (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP-Net [9]</td>
<td>6.61</td>
<td>NA</td>
</tr>
<tr>
<td>LoopReg [10]</td>
<td>9.12</td>
<td>NA</td>
</tr>
<tr>
<td>Fit to input</td>
<td>16.15</td>
<td>26.09</td>
</tr>
<tr>
<td>PHOSA [88]</td>
<td>13.73</td>
<td>34.73</td>
</tr>
<tr>
<td>Ours</td>
<td>4.99</td>
<td>21.20</td>
</tr>
</tbody>
</table>

Table 2. We compare our method to obtain SMPL and object fits with IP-Net, LoopReg and PHOSA. We also show that directly fitting SMPL and object meshes to the input leads to sub-optimal performance. Our method not only obtains better fits, but unlike LoopReg and IP-Net, we can also fit the object.

Object orientation prediction. To train the network \( f_\phi^b \), we use the point feature \( F_j \) of a sampled query point \( p_j \) to predict the global object orientation \( a_j = f_\phi^b(F_j) \). We jointly train \( f_\phi^b \) with standard L2 loss. We find that points far away from the object surface are unreliable in predicting the object orientation. Hence we only apply this loss to points that are close to the object, i.e. the GT \( u^b_j \) < \( c \). Since we have the GT object fit, we obtain the GT orientation by running PCA on the object mesh vertices and use the 3 principal axes in \( \mathbb{R}^9 \).

5.3. Why can’t existing human registration approaches be extended to our setting?

There are no direct baselines that can jointly track humans, objects, and contacts from multi-view input. There are works [9, 10] that pursue similar ideas of predicting correspondences and fitting SMPL to the human point cloud. In this subsection we explore their suitability in our setting.

Comparison with IPNet [9] IPNet takes as input a human point cloud and predicts an implicit reconstruction of the human and sparse correspondences to the SMPL model, which enables its fitting to the implicit reconstruction. This approach has three major disadvantages. First, querying occupancies for a \( 128^3 \) grid to obtain implicit reconstruction is expensive. Second, it predicts occupancies which requires water-tight surfaces. And third, running traditional Marching Cubes makes occupancy prediction non-differentiable w.r.t. SMPL fitting. Our formulation in Eqs. (2) and (3) alleviates these problems as we can fit SMPL by only querying \( N = 30k \) points instead of \( 128^3 \) \((\sim 2M)\) points. Since we use unsigned distance prediction, our method can work with non-water tight surfaces. We can also fit SMPL directly to unsigned distance predictions, thus removing the requirement for Marching Cubes. We compare our approach with IP-Net [9] (trained on our dataset) in Tab. 2 and show that we obtain better performance than IP-Net at much lower cost (30k (ours) vs. \( \sim 2M \) (IP-Net) query points and no Marching Cubes). This shows that our formulation is superior than IPNet even for human registration. We can additionally handle objects and interactions. Qualitative comparisons are given in the supplementary material.

Comparison with LoopReg [10] LoopReg fits SMPL to the input point cloud by explicitly predicting correspondences. We find the idea interesting and use their diffused
SMPL formulation in our method. LoopReg is, however, not directly applicable in our setting as it assumes a noise free and complete human point cloud. When the point cloud is incomplete due to occlusions, no correspondences are predicted for missing parts. Since LoopReg can only use surface points for fitting, this makes registration inaccurate. BEHAVE handles this case by using distances to the SMPL surface (Eq. (3)) predicted for each of the sampled query points to fit the body model, thus allowing the use of non-surface points for fitting. This is important as the Kinect point cloud is noisy. We outperform LoopReg [10] (trained on our dataset) and show (Tab. 2) that our formulation is robust to missing parts and noisy input.

5.4. Importance of contacts

In this experiment we show that our network predicted contacts are key for physically plausible tracking. Even though quantitative difference is not significant (Tab. 3), it can be seen in Fig. 5 that without contact information, the human and the object do not lock into the correct location. Hence, we notice unnatural results like floating objects. Using our contact prediction alleviates such issues. We encourage the readers to see our supplementary document for detailed discussion regarding limitations and future work with BEHAVE.

6. Conclusions

We have presented BEHAVE, the first methodology to jointly track humans, objects and explicit contacts in natural environments. By introducing neural networks to predict correspondences to a 3D human body model along with unsigned distance fields defined over human and object surfaces, we are able to accurately model human-object contacts. We further integrate such neural predictions into a proposed joint registration method resulting in the robust 3D tracking of human-object interactions. Along with our proposed method we also provide BEHAVE, the largest dataset of RGBD sequences and annotated humans, objects, and contacts to date. BEHAVE dataset is the first benchmark for the part of the research community interested in modelling human-object interactions. We propose real-world challenges like reconstructing humans and object from a single RGB image, tracking human-object interactions from multiple and single-view RGB(D) input, pose estimation etc. Our dataset together with our code is released in order to stimulate future research in this important emerging domain.

Acknowledgements Special thanks to RVH team members [4], and reviewers, their feedback helped improve the manuscript. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans), German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A and ERC Consolidator Grant 4DRepLy (770784). Gerard Pons-Moll is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.
References


[83] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a contact potential field to model the hand-object interaction. In ICCV, 2021. 2


[90] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In International Conference on 3D Vision (3DV), Nov. 2020. 2