

Visually plausible human-object interaction capture from wearable sensors

Vladimir Guzov^{1,2}, Torsten Sattler³, and Gerard Pons-Moll^{1,2}

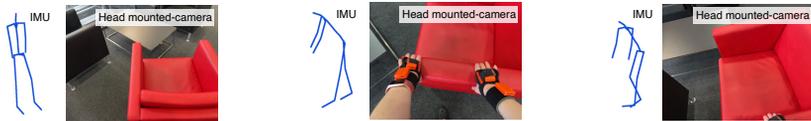
¹ University of Tübingen, Germany

{vladimir.guzov,gerard.pons-moll}@uni-tuebingen.de

² Max Planck Institute for Informatics, Saarland Informatics Campus, Germany

³ Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic

torsten.sattler@cvut.cz



Real-world interaction captured with wearable sensors



HOPS translates real-world interaction to the virtual world

Figure 1. We propose HOPS – a method of capturing human-object interactions using only a wearable camera and inertial sensors. Combining visual localization cues and our novel interaction model, HOPS tracks the human and the object pose even when it is not visible. Since our method is inherently dynamic, we urge readers to view our supplementary video for animated results.

Abstract. In everyday lives, humans naturally modify the surrounding environment through interactions, *e.g.*, moving a chair to sit on it. To reproduce such interactions in virtual spaces (*e.g.*, metaverse), we need to be able to capture and model them, including changes in the scene geometry, ideally from ego-centric input alone (head camera and body-worn inertial sensors). This is an extremely hard problem, especially since the object/scene might not be visible from the head camera (*e.g.*, a human not looking at a chair while sitting down, or not looking at the door handle while opening a door). In this paper, we present HOPS, the first method to capture interactions such as dragging objects and opening doors from ego-centric data alone. Central to our method is reasoning about human-object interactions, allowing to track objects even when they are not visible from the head camera. HOPS localizes and registers both the human and the dynamic object in a pre-scanned static scene. HOPS is an important first step towards advanced AR/VR applications based on immersive virtual universes, and can provide human-centric training data to teach machines to interact with their surroundings. The supplementary video, data, and code will be available on our project page at <http://virtualhumans.mpi-inf.mpg.de/hops/>.

1 Introduction

Current XR applications provide limited immersion as interaction with the virtual world either requires a controller, gloves, or other input device or is restricted to simply walking through the scene. At the same time, imagine the immersion that could be reached if it was possible to interact with the virtual world through interacting with objects in the real world. For example, opening a door in the real world would open a portal into another dimension in a virtual reality application, or moving a piece of furniture would solve a puzzle in an AR adventure game. Imagine the possibilities collaborative settings open up, where one person can virtually visit another space and observe another person interacting with their environment (see Fig. 2).

Common to the examples above is that they require capturing the interaction between a human and objects in the environment. Naturally, one could use an external camera system to identify and track both and this case has been covered in the literature [46, 47], including tracking objects within SLAM systems [38, 50]. However, expecting non-experts to buy, mount, and calibrate a multi-camera system while ensuring that there are no blind-spots, e.g., due to a human occluding the door while opening it (as in Fig. 7, seq. 4), seems very unrealistic and a deterrent to such applications becoming widely adopted. In order to be both scalable and to limit the efforts required by a user to setup the system, only wearable sensors such as cameras, IMU sensors, etc. should be used. Prior work showed that such a setup allows to precisely localize a human in a known scene and to capture their body’s motion [17]. However, this prior work assumes that the scene is static and is not capable of modeling changes in a scene caused by humans interacting with objects.

To the best of our knowledge, this paper is the first attempt to address a novel extremely challenging problem: *jointly tracking and localising the human and the dynamic scene changes resulting from the interaction, from wearable sensors alone*. Since the objects in the scene are often not visible in the head camera during the interaction, this might appear as an impossible task. The human might not look at the object during the interaction, e.g., while adjusting the position of a chair while sitting down. Or the object might appear static to the camera, e.g., a table might fill the human’s field of view while dragging it through the scene, creating the impression that neither table nor human move (see Fig. 3). It thus becomes necessary to include contact information, e.g., provided by pressure sensors.

Certainly, a traditional visual tracker is not sufficient, and metrically accurate tracking is nearly impossible in the absence of visual cues. However, many applications only require *visually plausible tracking* instead. That is obtaining a motion which looks qualitatively similar to the original motion, while satisfying physical and contact interaction constraints. Real humans are capable of such tracking of objects without permanently looking at them because contacts with the object give us a strong cue of the object location relative to us. We also have a mental model of the degrees of freedom of objects (e.g, a door can only move along a hinge joint, or a couch can only move on the XY plane unless it is lifted).

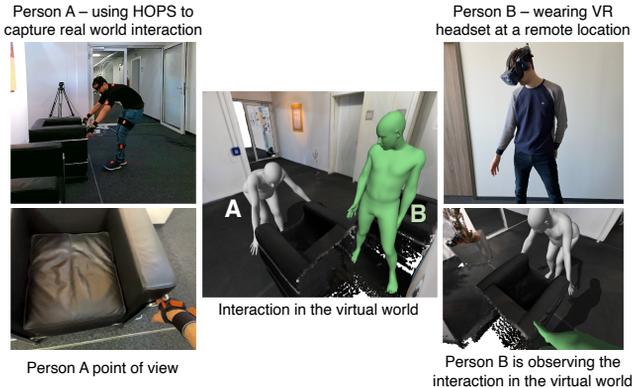


Fig. 2: **HOPS potential application.** One person can interact and change the real scene while the other can observe this interaction from a remote location in the virtual world.

These observations motivate the design of our novel method, which we name *HOPS – Human-Object Positioning System*. HOPS is the first method to localize and estimate the pose of the human and the object the person interacts with, within a pre-scanned 3D scene using IMUs and a head-mounted RGB camera. HOPS integrates several cues in a sequential optimization framework with interaction reasoning. For some frames, the person can be reliably localized in the scene with visual localization from the head camera. Such estimates are integrated jointly with IMU-based full body pose estimates. Similarly for some frames, the dynamic object (which can be in a different location relative to the static scan) is visually localized within the 3D scene from the head camera. Although this is only possible when the person is sufficiently far away from the object, before and after the interaction, such object and person localization estimates serve as *anchor points*. When the person interacts and *contacts the object*, person and visual object localization are typically no longer possible, because there are not enough reliable cues (as in Fig. 3). Hence, during contact interaction, we track the human using IMUs cues only. We track the object by enforcing that it moves coherently along with the human satisfying contacts and the degrees of freedom of the object – we parameterize the object motion to satisfy physical constraints (e.g., a door can only move along a hinge joint) [22]. All aforementioned cues are integrated to obtain qualitative human-object motion which is coherent with the anchor points, contacts and physical constraints during interaction.

We see our approach as step towards embodied capture of human-scene interactions, which can open a wide range of immersive AR and VR applications, for example, collaborative interaction with virtual world through modifying real-world objects. Furthermore, our approach can be used to record training data to teach robots and autonomous agents to interact with the world.

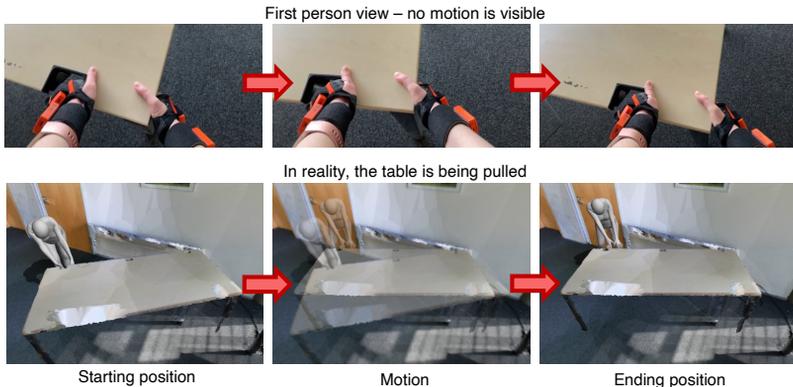


Fig. 3: **Body-mounted capturing challenges.** During the ego-centric camera capturing, the object might appear static to the camera, even if the motion is happening in reality. HOPS aims to overcome such problems and still track the object by using body motion then the visual data is not reliable.

In summary, our contributions are the following: **1)** We address a challenging new problem with many relevant applications: estimating human and object motion within a global 3D scene, all from ego-centric data alone. **2)** We propose HOPS, an optimization based method which combines visual localization, signal from wearable IMUs on the body, and reasoning about human-object contacts and physical constraints. **3)** We will release code and data to facilitate further research in this new direction.

2 Related Work

Human-object interaction: Currently, most of the methods work with external cameras to record human-object interactions. Methods to capture the full body pose are using external cameras and mostly static scenes [19] or not using the scene context at all [55, 58, 64]. RigidFusion [56] tracks objects using an external RGBD sensor. There are methods that work with first person view footage, however they are mostly studying the upper part of the body, *e.g.*, hand-object pose estimation [13, 26, 29, 33], and are mostly limited to static cameras. Our method works with body-mounted sensors and a moving camera while capturing the full-body pose and object position simultaneously.

Embodied research: Body-mounted sensor setups are heavily used to solve various tasks: activity recognition methods like [4, 9, 15, 32, 37, 61] use ego-centric camera setups with a camera looking towards the body. However they typically concentrate on capturing the upper body. Many full-body capturing methods [36, 49, 57] are working with similar head-mounted setups, but because the cameras are looking on the person wearing them rather than outwards, these methods do not take the environment around the subject into consideration. There are

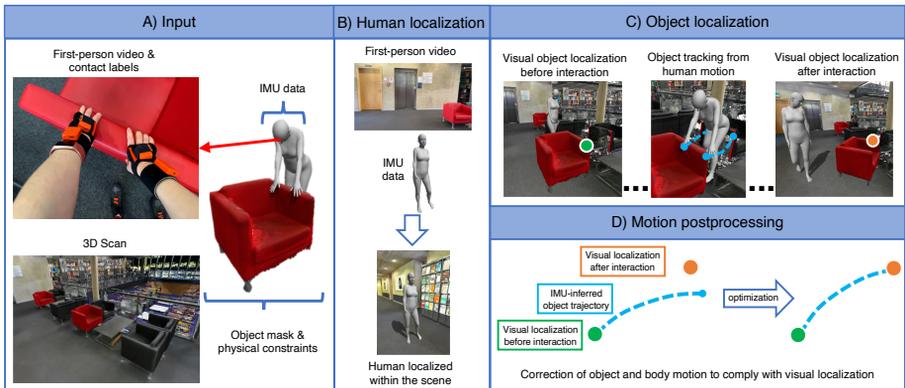


Fig. 4: **Overview.** (A) HOPS uses RGB video from a head mounted camera, human body localization data from IMUs, contact labels, a pre-scanned scene and object physics constraints as well as object masks for RGB video and the scene 3D scan. Our method performs a visually plausible human-object interaction modeling by localizing human (B), inferring object position from body motion and visual cues (C) and correcting the object and body motion parameters (D) to match the visual localization results using the bending energy (Sec. 3) optimization.

methods that work with an outwards-facing camera [24, 62, 63], however they do not use any additional sensors to capture the human body pose, instead predicting it using motion priors. This results in body poses far from the ground truth. Most related to our method, HPS [17] captures human motion by using body-mounted IMU sensors and a head-mounted camera looking outwards to initialize and further correct the location of the subject within a pre-built 3D scan of the environment. Our approach essentially extends [17] by not only tracking the human pose but also an object the person interacts with. Whereas HPS is restricted to static environments and cannot model scene changes caused by human-object interactions, HOPS removes these restrictions.

Visual localization: Visual localization algorithms aim to estimate the pose of a camera in a known environment. Current state-of-the-art approaches for accurate camera pose estimation are based on 2D-3D matches between pixels in the camera image and 3D scene points. These 2D-3D matches are either estimated based on matching local features [21, 28, 31, 39, 41, 42] or by regressing a 3D point coordinate for each pixel [6, 7, 10, 12, 44]⁴. A recent line of work focuses on (benchmarking) the robustness of localization algorithms in changing scenes [12, 23, 48, 53], *i.e.*, to illumination, weather, and seasonal changes as well as to changes caused by human actions (rearranging furniture, *etc.*). These approaches assume that a large-enough part of the scene remains static

⁴ We refer the interested reader to [5] for a discussion and comparison of both types of approaches.

and is observed by the camera to facilitate pose estimation. The second assumption is violated in our scenario and we use IMU-based human pose tracking to bridge gaps where visual localization fails. We use a state-of-the-art localization pipeline [39, 40]. For convenience, we use the same technique for object pose estimation. Using a dedicated object pose estimation algorithm such as [27, 43, 54, 59] should further improve performance. Note that the main contribution of HOPS, *i.e.*, jointly reasoning about human and object poses, is not tied to any particular localization algorithm.

3 Mathematical background

Our method is built on an efficient combination of data coming from the IMUs and head-mounted camera. However, these two sources of data can often give different predictions for the human and object positions. For example, IMU pose estimation is consistent across frames, but accumulates drift over time which cannot be fixed as IMU sensors cannot track the global position in the scene. In contrast, camera localizations do not drift, but the resulting poses are noisy. Therefore it is crucial to have a way to efficiently combine these two data sources, resulting in drift-free and smooth trajectories both for the object and body motions. Our solution is, using the IMU estimations, to build a trajectory and correct it with camera localization estimates, ignoring the noise and keeping the motion trajectory smooth and realistic.

In this section we describe a general mathematical tool that allows us to do so: with this tool one can adapt any trajectory to a set of “control points” – a set of points which trajectory should lean to intersect, while deforming the least. The key idea here is to interpret deviations in a trajectory from the original as incurring a potential energy, analogously to flexing a tape. In the context of our task, this provides a trade-off between remaining close to the captured human motion (initial IMU estimate) and satisfying the visual localization constraints. The intuition is to preserve the local characteristics (first and second derivatives, curvature) of the reference trajectory and hence its naturalness, while satisfying constraints. This is useful since IMU trajectories are locally accurate, but globally subject to drift.

Position trajectory deformation. Despite the 3D nature of our method, here we are focusing on the 2D trajectories. In practice, we correct only the x and y components in the IMU trajectory and leave z component unchanged. Given a trajectory described as a curve $\mathbf{l}(t) = (x(t), y(t))$, $t = [t_{\text{start}}, t_{\text{end}}]$, and a list of K control points $\mathbf{p} = \{\mathbf{p}_i = (x_i, y_i)\}_{i=1}^K$ (constraints) at times $t_{\text{start}} \leq t_1, \dots, t_K \leq t_{\text{end}}$, we minimize the trajectory bending energy E_{tr} and the distance to the control points as follows:

$$F_{tr}(\mathbf{l}, \mathbf{p}) = \arg \min_{\hat{\mathbf{l}}} \left(\sum_{i=1}^K (\|\hat{\mathbf{l}}(t_i) - \mathbf{p}_i\|_2) + \lambda E_{tr}(\hat{\mathbf{l}}, \mathbf{l}) \right), \quad (1)$$

where λ is the tape rigidity coefficient (Fig. 5, A). The bending energy penalizes deviations to the initial curve \mathbf{l}

$$E_{tr}(\hat{\mathbf{l}}, \mathbf{l}) = \int_{t_{\text{start}}}^{t_{\text{end}}} \left(\frac{d\hat{\alpha}(t)}{dt} - \frac{d\alpha(t)}{dt} \right) dt; \quad (2)$$

where

$$\hat{\alpha}(t) = \arctan \left(\frac{d\hat{y}}{dt}, \frac{d\hat{x}}{dt} \right), \quad \alpha(t) = \arctan \left(\frac{dy}{dt}, \frac{dx}{dt} \right).$$

Here, α is the angle of the curve tangent vector and the energy penalizes deviations of the angle’s temporal derivative. We can interpret it as forcing curvature preservation.

Body pose trajectory deformation. For pose θ we propose a similar solution, except that the bending energy

$$E_{\theta}(\hat{\theta}, \theta) = \int_t \left\| \frac{d\hat{\theta}(t)}{dt} - \frac{d\theta(t)}{dt} \right\|_2 \quad (3)$$

depends only on the first temporal derivative (for simplicity in high dimensions), instead of the second derivative as in Eq. (2). The control points are target positions \mathbf{p} for certain vertices $\mathbf{j} = \{j_i\}$ of the SMPL model at times $\{t_i\}_{i=1}^K$. We find the trajectory by minimizing

$$F_{\theta}(\theta, \gamma, \mathbf{p}, \mathbf{j}) = \arg \min_{\hat{\theta}, \hat{\gamma}} \left(\lambda(E_{\theta}(\hat{\theta}, \theta) + E_{tr}(\hat{\gamma}, \gamma)) + \sum_{i=1}^K (\| (M_{j_i}(\hat{\theta}_{t_i}, \hat{\gamma}_{t_i}))_{x,y} - \mathbf{p}_i \|_2) \right), \quad (4)$$

where $(\cdot)_{x,y}$ means that only the x - and y -axis components are used, and $M_{j_i}(\cdot) \in \mathbb{R}^3$ is the j_i -th SMPL vertex. All temporal derivatives are approximated with finite differences. Fig. 5, B shows qualitative results of optimization.

4 Method

We aim to estimate the 3D body and object poses during interaction from body-mounted sensors (head camera and body-worn IMUs), while localizing both within a given 3D scene – a static 3D scan of the scene is an input to our method as in HPS [17]. Formally, our task is to estimate the human body pose $\theta = \{\theta_t\}$, translation parameters $\gamma = \{\gamma_t\}$, and object position $\mathbf{O} = \{\mathbf{O}_t\}$ at time $t \in [t_{\text{start}}, t_{\text{end}}]$. For visually plausible tracking, we need to satisfy the following (soft) constraints:

1. The object should be close to the object visual localization \mathbf{O}_t estimates at discrete *anchor* keyframes $k \in T^O \subseteq [t_{\text{start}}, t_{\text{end}}]$, when visual localization is reliable. Also, the human pose should match the visual localization of the head camera \mathbf{C}_t for reliable intervals $t \in T^C$.

2. During the interaction intervals $T^{\text{int}} = \{[t_s^i, t_e^i]\}_{i=1}^K \subset [t_{\text{start}}, t_{\text{end}}]$, contact locations derived from the human body $c^{\text{body}}(\boldsymbol{\theta}_t, \boldsymbol{\gamma}_t)$ should match the ones derived from object location $c^{\text{obj}}(\mathbf{O}_t)$.
3. Body parameters $\boldsymbol{\theta}_t, \boldsymbol{\gamma}_t$ (pose and translation of the SMPL [30] model) should remain close to the IMU estimates $\{\boldsymbol{\theta}_t^I, \boldsymbol{\gamma}_t^I\}$, registered to the scene.

To solve such a complex task, we divide it into several subproblems: we perform human body localization (Sec. 4.1) and object localization (Sec. 4.2) in the given scene and then refine the result to ensure human and object pose consistency (Sec. 4.3). Fig. 4 illustrates our method.

Since the focus of this paper is visually plausible tracking and interaction reasoning, we concentrate on interactions that involve the full body and large objects whose degrees of freedom are constrained by the environment (e.g., a door can only move along the hinge joint). We decided to assume a mask of the object and contact labels as given. Our choice is motivated by our belief that such data can be obtained automatically during the recording in the near future, although right now it requires setting up additional hardware or developing additional algorithms, therefore we leave this as a future work.

We obtain the mask using a semi-automatic method [45], which we use for both the scene scan images and for the head camera images⁵ – given the rapid progress in semantic segmentation, we do not see using a semi-automatic method as an important limitation. Contact labels are given in form of time stamps of when the interaction starts and ends, as well as an indication which of the hands are used during the interaction (left, right, or both). Such labels would be possible to capture with hand touch sensors, or from visual cues if the hands are in the field of view, but we leave this for the future work.

4.1 Human body localization

IMU-based pose estimation. Current IMU-based pose estimation solutions, such as XSens [35] used in our setup, allow us to get a continuous human pose estimate using a wearable and small IMU sensors. Xsens system consists of 17 IMUs attached to the body with velcro-straps. Using proprietary algorithm based on the Kalman filter and kinematic body, the system provides human pose $\tilde{\boldsymbol{\theta}}_t^I$ and translation estimates $\tilde{\boldsymbol{\gamma}}_t^I$ in its own coordinate frame. While these estimates can be considered accurate, the system has two major flaws: 1) motions are not registered within the 3D scene and 2) these estimates accumulate significant drift over time, especially in the global body position and orientation components.

Head-mounted camera localization. For every camera frame, we get the initial head pose estimates using hierarchical localization algorithm [39], which operates on RGB images. Given the dataset of images with known position in the pre-scanned scene, the algorithm establishes correspondences between those images and frames coming from the camera and finds the position of the the head camera $\tilde{\mathbf{C}}_t$ by minimizing the reprojection loss.

⁵ As in [52], we thus assume knowledge about which object changes.

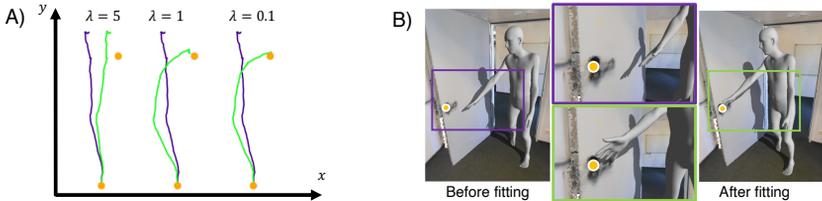


Fig. 5: **Trajectory and pose parameters fitting with bending energy.** A) bending trajectories with F_{tr} using different rigidity coefficients λ , purple marks the original trajectory, green marks the result; B) Visualization of the SMPL model after fitting parameters with F_{θ} to minimize the distance between the hand and the control point. Orange dots denote control points.

Combining camera and IMU. Inspired by HPS [17], we use a data from the head-mounted camera to register the body within the scene and correct the IMU drift. Our body localization module uses these head camera locations $\tilde{\mathbf{C}}_t$ and initial IMU estimates $(\tilde{\theta}_t^I, \tilde{\gamma}_t^I)$ to derive the body position within the scene and correct body and camera positions using the bending energy minimization introduced in Sec. 3. As a result, we get improved camera locations \mathbf{C}_t and human pose parameters (θ_t^I, γ_t^I) registered within the scene. Please see the supplementary for more detailed description of the algorithm.

4.2 Object localization

Robust visual object localization for anchor keyframes. In contrast to localizing the human head camera which is in permanent motion, the objects we consider are static and can only move during the interaction with the human. We use this observation and restrict object visual localization to detecting a few anchor keyframes, which will serve as control points during interaction reasoning here and in Sec. 4.3.

After localizing the camera $\mathbf{C}_t \in SE(3)$ for every frame $t \in T^C$, $T^C \subset \mathbb{R}$, we compute the object locations $\mathbf{O}_t^C \in SE(3)$ relative to the camera. Here, we use the same camera localization method from Sec. 4.1, but use only keypoints inside the object mask (both for the head camera frame, and the database images). The position and orientation of the object \mathbf{O}_t in world coordinates is obtained as: $\mathbf{O}_t = \mathbf{C}_t(\mathbf{O}_t^C)^{-1}$. Since objects only move when the human interacts with them, we leverage temporal information to accurately locate them. We split object positions in $K + 1$ groups, where K is the expected number of dynamic interaction intervals. Each group contains object positions detected before interaction interval $K + 1$ and after K . To filter the noise, for each group, we run the clustering algorithm DBSCAN [14] on the translation part of every location in the group and leave only the largest cluster. We compute the mean position per group, and sort them temporally, which gives us $K + 1$ reliable positions $\{\mathbf{O}_t\}_{t=1}^{K+1}$.

Tracking objects from human motion, even when they are not visible. The set of reliable object localization estimates $\{\mathbf{O}_t\}_{t=1}^{K+1}$ from the previous paragraph is limited – the object should be visible and contain distinct features on the surface. As we are dealing with real-world interactions, we cannot guarantee that these conditions are met during the interaction. The key idea here is to leverage the human motion and the physical constraints of object motion to address an otherwise impossible task: tracking objects that are not visible. Every object type has its own physical behaviour and restrictions – *e.g.* some of the objects can be carried everywhere, but some of them are fixed or cannot be lifted and can only be dragged. Therefore the solution is class-specific: here, we describe our approach for objects moved by hands along the floor plane (chairs, tables, *etc.*), or fixed to a vertical hinge (doors), but the method can be extended to the other object classes with different physical restrictions (drawers, sliding doors, *etc.*).

Human trajectory. For each hand marked to be in contact at the start of the interval (input to our method), we find the closest point to the hand on the object $\mathbf{c}^i, i \in \{L, R\}$ (left, right) and mark it as a contact. Then we compute the offset we need to apply to the human root joint such that hands \mathbf{h}^i are in contact by $r_{\text{offset}} = \frac{1}{2} \sum_{i \in \{L, R\}} (\mathbf{c}^i - \mathbf{h}^i)$ in case of two hands or simply $r_{\text{offset}} = (\mathbf{c}^i - \mathbf{h}^i)$ in case of one hand. The desired SMPL root location r_{root} at the start of interaction is therefore $\mathbf{p}_1 = r_{\text{root}} + r_{\text{offset}}$. Hence, we deform the full trajectory using the method described in Sec. 3 with a single control point $\hat{\gamma}^I = F_{tr}(\gamma^I, \{\mathbf{p}_1\})$.

Tracking objects without seeing them. For *objects dragged on the floor* with two hands, we set the object translation $\Delta \mathbf{t}^O = \Delta \mathbf{h}^R$ to the translation of the right hand $\Delta \mathbf{h}^R$ (the choice of left or right is mathematically equivalent). The change of rotation $\Delta \mathbf{R}^O$ is determined by the angle change between left and right hand joints. If the object is dragged with one hand, then $\Delta \mathbf{t}^O = \Delta \mathbf{h}^X$, where $X = L$ or R depending on the hand in contact. The rotation cannot be determined from 1 contact point and remains unchanged, however it will be interpolated if visual localization is available (Sec. 4.3). For the *objects attached to vertical hinge* we restrict inference to a planar rotation $\mathbf{R}(\alpha)$ along the hinge with angle α . This is done by computing the angle between previous and current hand to hinge joint vectors $\Delta \alpha = \widehat{(\mathbf{v}_i, \mathbf{v}_{i-1})}, \mathbf{v}_i = \mathbf{h}_i^X - \mathbf{j}$, where \mathbf{j} is hinge joint position on the floor, and $X = L$ or R depending on the hand in contact.

4.3 Human-object interaction refinement: Inferring human and object motion coherently with object localization

With the previously explained human-object reasoning method, we can track objects, but the human motion might drift during the interaction (since motion relies on IMU). Here, we can leverage the object localized at the end of the interaction (note that only one good visual localization frame is needed to localize the object after interaction) to self-correct both the human and object trajectories.

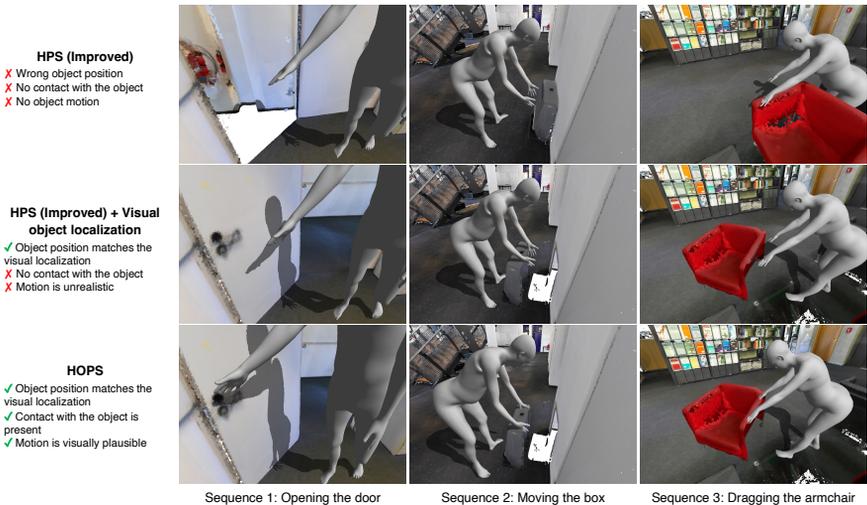


Fig. 6: **Effect of different components of HOPS.** Here, the different interactions are shown. While other methods are either losing contact with the object, or does not track the object at all, HOPS ensures that the object and the body are both correctly positioned and interacting in a realistic way.

Without loss of generality, we explain our method for a single interaction interval. Let \mathbf{O}_s and \mathbf{O}_e be the object visual localizations at the start and end of the interaction, obtained using the method in Sec. 4.2. Given the starting contact positions \mathbf{c}_s from Sec. 4.2 (to simplify notation we use \mathbf{c}_s to denote the contact location of the left, right hand or both depending on the situation), we determine the contact points at the end of the interaction as $\mathbf{c}_e = \mathbf{O}_e \mathbf{O}_s^{-1} \mathbf{c}_s$.

Next, we deform the hand trajectories $\mathbf{h} = \{c^{\text{body}}(\boldsymbol{\theta}_t, \boldsymbol{\gamma}_t)\}_{t=s}^e$ to fit start and end control points $\{\mathbf{c}_s, \mathbf{c}_e\}$ using again the bending energy $\hat{\mathbf{h}} = F_{tr}(\mathbf{h}, \{\mathbf{c}_s, \mathbf{c}_e\})$ (Fig. 4, D). If the object has a hinge, hand trajectories $\hat{\mathbf{h}}$ are additionally projected to the subspace of possible trajectories for this object. The resulting pose trajectory is then obtained by deforming the pose estimates to fit the newly found hand trajectories as $\boldsymbol{\theta}, \boldsymbol{\gamma} = F_{\theta}(\boldsymbol{\theta}^I, \hat{\boldsymbol{\gamma}}^I, \hat{\mathbf{h}})$.

5 Experiments

Our approach is, to the best of our knowledge, the first attempt to solve a novel and difficult problem of tracking human object interactions and capturing dynamic scene changes from wearable sensors alone. Dealing with an under-constrained task (modeling interactions that are not visible to the camera) our method aims to produce visually plausible motion, rather than ground truth human motion and interactions.

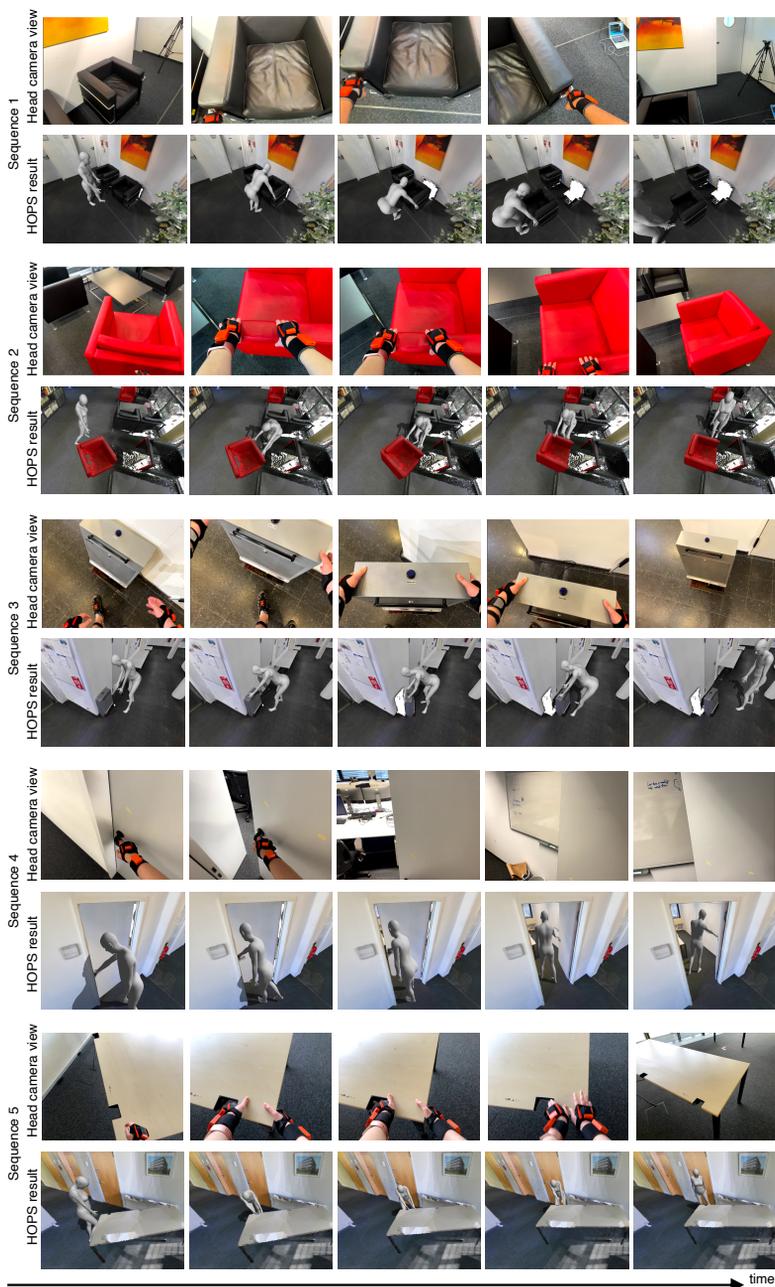


Fig. 7: **Qualitative results of our method.** To visualize the motion, several frames of the person approaching and interacting with the object are shown next to each other. Please see the supplementary video for more details and results.

Hence, qualitative evaluations are more meaningful in this setup, and we consciously focus on qualitative results in our work (see supp. video).

As we are facing the lack of evaluation data to prove the effectiveness of HOPS, we captured our own dataset, consisting of 30+ minutes of footage (more than 50K frames) from the head-mounted camera coupled with segmentations of the target object for every 10th frame, and inertial data from the on-body IMU sensors, which we will make publicly available. All subjects explicitly agreed to being recorded.

Fig. 7 shows HOPS results for some of the sequences. As we are working with moving humans and objects most of the sequences are best seen in motion. We therefore urge the reader to see the supplementary video for more results.

Comparisons. To prove the usefulness of being able to adapt to dynamic environment, we compared our method to the baseline human-only localization method from Sec. 4.1, which is essentially an improved version of HPS [17] featuring the revised localization pipeline and minor adaptations to improve stability. We refer to it as **HPS (Improved)**. To show the advantage of using camera and IMU information jointly, we designed the version of the algorithm which does not use body motion to infer the object position and only relies on the visual object localization, which is a combination on the improved HPS algorithm and a visual object localization. As it is not possible to visually localize the object during the interaction, the algorithm linearly interpolates the object motion between starting and ending positions. We refer to this method as **HPS (Improved) + Visual object localization**. Note that the body localization part (Sec. 4.1) still uses information from both the camera and the IMUs, but the human-object interaction refinement step 4.3 is not applied. The results are presented in Fig. 6 and supplementary video. We observe that, compared to the HPS baseline, “HPS (Improved) + Visual object localization” method improves performance, however, as it is only possible to reliably localize the object before and after the interaction, the interpolation between those positions does not reflect the real motion (see supp. video). A lack of pose correction also leads to inaccurate contact position. HOPS benefits from both visual and motion clues resulting in the most accurate representation of a dynamic scene.

6 Conclusions and Future Work

In this paper, we have considered the novel and challenging problem of modeling the interaction between a human and an object in the scene from ego-centric data alone while simultaneously estimating the poses of both in a known scene. Our proposed approach, HOPS, achieves this goal by combining head-mounted camera-based visual localization of the human and the object with human body pose estimation from IMU data. HOPS reasons about the human-object interaction by minimizing an energy function that contains terms corresponding to contact and physical constraints. Our experiments have shown that HOPS allows us to realistically model these interactions.

Limitations and future work. To the best of our knowledge, our approach is the first to tackle the problem of human-object interaction from egocentric data while positioning both in a scene. It thus represents a first and important step towards novel immersive applications, *e.g.*, by enhancing AR/VR experiences via interaction, and thus towards the recently often-mentioned Metaverse [1]. Yet, many challenges still need to be overcome: currently, our approach focuses on a single object movement which is segmented semi-automatically. Ultimately, the goal should be tracking every dynamic motion happening in the scene, without the need to guide the segmentation. Using learned human motion priors, it would be also interesting to capture interactions using a reduced set of body-worn IMUs [20, 51, 60]. Further, automatically anticipating the time and location of contacts, as well as introducing learned human-object interaction priors to handle more complicated activities and avoid interpenetration are important directions for future work. To promote future research in this new exciting direction, we will release our datasets and code.

Ethical implications. As with any approach that analyzes human behavior, there is potential for misuse. This includes, but is not limited to, surveillance, building profiles of individual users based on their interactions (and selling that data, *e.g.*, for advertisement purposes), and identifying individuals based on characteristic interactions and movements. Our approach is based on ego-centric data, which, in combination with on-device processing, offers a user some level of control over what data is shared with other parties. Further, one could report generic rather than individual body poses to make it harder to prevent identifying individuals based on their movement. Still, it is not possible to prevent user surveillance or data aggregation as a post-processing step in later stages of an applications. We believe that these issues would best be addressed by law makers.

Acknowledgments. We thank Bharat Bhatnagar, Verica Lazova, Ilya Petrov and Garvita Tiwari for their help and feedback. This work is partly funded by the DFG - 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), the EU Horizon 2020 project RICAIP (grant agreement No.857306), and the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15_003/0000468). Gerard Pons-Moll is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 - Project number 390727645.

Supplementary — Visually plausible human-object interaction capture from wearable sensors

Abstract. In this supplementary material, we first (Sec. 7) provide more details about our method. In Sec. 8, we provide implementation details. In Sec. 9, we provide additional experiments and details about our experimental setup. Our project page also includes a supplementary video illustrating our approach, comparisons and qualitative results for multiple sequences. In addition, the video presents failure cases and points out directions for future work. Our work considers the problem of modeling the interaction between a human and an object in the scene. Such interactions are dynamic by nature. The results of our approach are thus best appreciated in motion and we strongly encourage the reader to look at the video.

7 Method details

7.1 Human body localization based on a head-mounted camera and IMUs

To effectively reason about object contact and fine-grained interactions, we need a good initialization of the human body pose first. For that, we need to register the IMU pose estimation within the scene coordinate system. To achieve so, we use a 5-step initialization method: **Step 1: Initial camera estimates.** To align IMU coordinate frames with the world and camera coordinate frames, we compute initial camera localization estimates at each frame $\tilde{\mathbf{C}}_t$. The localization method is similar to the one used in HPS [17] with several modifications. For each camera frame time t , we detect keypoints [11] and match them with keypoints with known 3D position obtained using the prescanned scene. This is done with the SuperGlue [40] matching network, which outputs matches $\{m_t^i\}_{i=1}^M$ along with their confidence scores $\{s_t^i\}_{i=1}^M$. To find the camera location, we minimize the reprojection loss of the matching 3D points w.r.t. the camera pose. We save the average matching confidence $s_t = \frac{1}{M} \sum_{i=1}^M s_t^i$ to reason about the localization precision in our pipeline.

Step 2: Finding the mapping between IMU, world and camera coordinate frames. IMU poses $\tilde{\theta}_t^I$ and translation estimates $\tilde{\gamma}_t^I$ obtained from the IMU mocap system [35] need to be transformed to the world coordinate frame (*i.e.*, the pre-scanned 3D scene). Specifically, we find transformation matrices from the global IMU coordinate frame “Z” to scene “W”) $\mathbf{X}^{WZ} \in SE(3)$ and local (camera “C” to IMU sensor “I”) $\mathbf{X}^{IC} \in SE(3)$ coordinate frames – see supp. Sec. 7.2 for details. With this we can obtain the absolute head camera pose estimated by the head IMU, as $\mathbf{H}_t^W = \mathbf{X}^{WZ} \mathbf{H}_t^Z \mathbf{X}^{IC}$, where $\mathbf{H}_t^Z \in SE(3)$ are the raw head IMU positions.

Step 3: Improving the camera estimates. We rerun the camera localization with the search space for 2D-3D matches restricted to a vicinity of the current local scene in the human field of view. To this end, we filter out matches based on the IMU head pose estimates \mathbf{H}_t^W . We render a depth map of the camera view as seen from the head IMU; with the depth map, we filter out all keypoints in the database images that are not visible in this virtual view. This drastically restricts the search space from the full building database to only the current view. Localizing the camera with only filtered keypoints results in camera estimates \mathbf{C}_t (and their corresponding scores s_t), which are significantly more accurate than the ones in Step 1. Hence, we also recompute \mathbf{X}^{WZ} and \mathbf{X}^{IC} with \mathbf{C}_t to increase the IMU-to-scene mapping precision.

Step 4: Correcting IMU drift with visual localization. Pose estimation inferred from the IMUs can drift over time. We solve this problem by minimally bending the trajectory of the head from the IMU pose estimation to comply with the head-mounted camera localizations. For that, we consider only reliable camera estimates \mathbf{C}_t with confidence $s_t \geq 0.5$, which are used as a set of control points in the bending energy minimization framework described in Sec. 3 in the paper:

$$\hat{\mathbf{r}}^H = F_{tr}(\mathbf{r}^H, \{\mathbf{C}_t | s_t \geq 0.5\}) , \quad (5)$$

where $\mathbf{r}^H = \{\mathbf{r}_t^H\}$ is the translation part of the head position \mathbf{H}_t^W . The corrected head trajectory and initial IMU pose estimates $\tilde{\boldsymbol{\theta}}_t^I, \tilde{\gamma}_t^I$ are used in the next step to produce the human body pose and translation parameters $\boldsymbol{\theta}_t^I, \gamma_t^I$.

Step 5: SMPL parameters adaptation to the new IMU trajectories. After correcting the head IMU trajectory with camera localization, we adapt SMPL parameters $\boldsymbol{\theta}_t^I, \gamma_t^I$ to fit the new trajectory. We initialize $\boldsymbol{\theta}_t^I, \gamma_t^I$ with $\tilde{\boldsymbol{\theta}}_t^I, \tilde{\gamma}_t^I$. We map the parameters from the global IMU coordinate frame “Z” to the scene coordinate frame “W” with \mathbf{X}^{WZ} . Translation part is mapped as $\gamma_t^I = \mathbf{X}^{WZ} \tilde{\gamma}_t^I$ and the body is rotated according to the rotation part of \mathbf{X}^{WZ} by modifying the first 3 values of $\boldsymbol{\theta}_t^I$ (which correspond to the global body rotation).

Next, we take the original trajectory \mathbf{r}^H and the refined trajectory $\hat{\mathbf{r}}^H$ and compute difference in rotation and translation. Our goal is to apply the same rotation and translation difference to the SMPL parameters $\boldsymbol{\theta}_t^I, \gamma_t^I$. To do so, we treat trajectories as a discrete set of points $\mathbf{r}^H = \{\mathbf{r}_i^H\}$. As a result, for each SMPL pose and translation parameter set $\boldsymbol{\theta}_i^I, \gamma_i^I$ there is a corresponding trajectory point \mathbf{r}_i^H (same for $\hat{\mathbf{r}}_i^H$). We project trajectories to the XY plane, zeroing out Z axis component (our experiments show that Z axis coordinate component usually does not require correction). For each SMPL pose index i we compute the trajectory pose and rotation angle change: $\Delta\gamma_i^I = \hat{\mathbf{r}}_i^H - \mathbf{r}_i^H$, $\Delta\alpha_i = (\widehat{v_i}, v_i), \hat{v}_i = \hat{\mathbf{r}}_i^H - \hat{\mathbf{r}}_{i-1}^H, v_i = \mathbf{r}_i^H - \mathbf{r}_{i-1}^H$. The translation parameters are updated as $\gamma_i^I = \tilde{\gamma}_i^I + \Delta\gamma_i^I$. The first 3 pose parameters $\boldsymbol{\theta}_i^I$ corresponding to the root SMPL joint are modified so that the root joint rotates by $\Delta\alpha_i$ around the Z axis. The resulting human pose parameters ($\boldsymbol{\theta}_i^I, \gamma_i^I$) are drift-free and registered within the scene.

7.2 IMU-Camera coordinate frames calibration.

As stated in supp. Sec. 7.1, Step 2, to use IMU pose estimations we need to find the mapping between the IMU and camera local and global coordinate frames. Using the notation introduced in supp. Sec. 7.1, we first find transformation matrices from the global IMU coordinate frame “Z” to scene “W” $\mathbf{X}^{WZ} \in SE(3)$. For that, we rigidly align a sequence of head IMU positions in 3D to the initial noisy camera positions (without considering the orientations). In our data, the Z axis is aligned to the gravity vector for both IMU frame “Z” and the scene frame “W”, so we project IMU and camera positions to the XY plane and only perform the alignment in the XY plane. To align IMU positions to noisy camera observations, we use RANSAC [16] algorithm, which features noise detection in the data. To synchronize camera and IMU timings we repeat RANSAC fitting with several time offsets, performing a grid search; the time synchronization is corrected manually if needed.

Next, we obtain transformation matrix between the local (camera “C” to IMU sensor “I”) $\mathbf{X}^{IC} \in SE(3)$ coordinate frames. For this, we select $K = 5$ camera positions $\{\tilde{\mathbf{C}}_k\}_{k=1}^K$ (transformations from local to global camera coordinate frame). We select the camera positions by measuring the distance between all the camera translations and the IMU head translations $\{\mathbf{X}^{WZ} \mathbf{p}_t^H\}$ mapped to the scene coordinate frame, where \mathbf{p}_t^H is the translation part of the raw head IMU position \mathbf{H}_t^Z . We select K camera positions with the smallest distance. We also select the corresponding head IMU positions $\{\mathbf{H}_k^Z\}_{k=1}^K$. For each $k = 1..K$, we compute the local camera to IMU transformation $\hat{\mathbf{X}}_k^{IC}$: $\hat{\mathbf{X}}_k^{IC} = \mathbf{H}_k^{Z^{-1}} (\mathbf{X}^{WZ})^{-1} \tilde{\mathbf{C}}_k$. To obtain the target \mathbf{X}^{IC} , we average all $\hat{\mathbf{X}}_k^{IC}$ with the following algorithm: As $\hat{\mathbf{X}}_k^{IC} \in SE(3) \forall k = 1..K$, each of them can be subdivided to rotation matrix $\hat{\mathbf{R}}_k^{IC} \in SO(3)$ and translation vector $\hat{\mathbf{t}}_k^{IC} \in \mathbb{R}^3$. For translation, we simply find an arithmetic average $\mathbf{t}^{IC} = \frac{1}{K} \sum_{k=1}^K \hat{\mathbf{t}}_k^{IC}$, for rotation \mathbf{R}^{IC} we use chordal L_2 mean [18]. Averaged rotation and translation $(\mathbf{R}^{IC}, \mathbf{t}^{IC})$ form the resulting transformation \mathbf{X}^{IC} .

SMPL parameters optimization details. To ensure smooth parameters change while optimizing SMPL parameters to fit the trajectory during human-object interaction refinement (Sec. 4.3 in the paper), we additionally allow w parameters before and after the interaction interval to change (w is a hyperparameter), therefore avoiding abrupt transition. Additionally, to stabilize motion we restrict the parameters change only to particular joints, namely hands, arms, shoulders and pelvis joints.

8 Implementation details

8.1 Capturing setup

We use a combination of Apple iPhone 12 with a head mount to capture first-person view and Xsens Awinda [35] IMU system to capture body motions. The iPhone 12 camera is capturing at a resolution of 1920×1440 , 30 FPS. While the

camera features automatic stabilization, we turn it off explicitly to get a better idea of the head motion. We use OpenCV [8] to get the camera intrinsics. Xsens Awinda consists of 17 IMU sensors attached to the body with velcro-straps and captures the body motion at 60 FPS. The camera and IMU systems record their data independently, for that reason the data need to be temporally and spatially synchronized (*cf.* supp. Sec. 7.2) afterwards.

Xsens Awinda outputs the data in form of skeleton joint angles which is not directly compatible to SMPL [30] pose parameters format. To convert between these formats we developed the following retargeting algorithm: we export the motion from Xsens internal format (MVN) into Autodesk FBX format and use Autodesk FBX Python SDK [2] to read and extract rotation for each joint in form of quaternions. We convert those quaternions to axis-angle representation used in SMPL. We map joint rotations from FBX skeleton to SMPL skeleton according to manually designed mapping. As a result, we can produce SMPL pose parameter vector by concatenating the mapped axis-angle joint rotations in the right order. While this method works well in most cases, we notice some pose artifacts during specific motions, *e.g.* the stomach is sometimes bent too much forward while the person leans forward to grab an object (*e.g.* supp. Fig. 8). This artifact appears in all of the methods we explored (HPS (Improved), HPS (Improved) + Visual object localization, HOPS) because the error happens on the data preprocessing stage and does not depend on the chosen method. One possible way to improve the quality of retargeting is to switch to fitting the SMPL model to chosen Xsens skeleton joints instead of copying joint rotations directly – we aim to explore this in the future work.

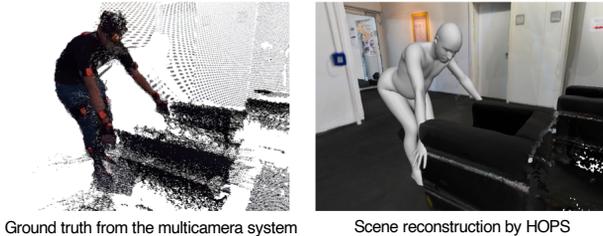
8.2 Algorithms implementation and performance

We implement our algorithms on Python using Pytorch [34] library for the visual localization pipeline and the bending energy optimization algorithm. For optimization of trajectories and pose parameters with bending energy (eq. 2 and 5 in the paper), we use Adam [25] optimizer with number of iterations, learning rate and rigidity coefficient λ acting as hyperparameters.

Performance-wise, the most computationally expensive part of our algorithm is the visual localization pipeline, requiring 8 seconds per frame to run on NVIDIA Q8000 GPU. Other parts of the algorithm does not add significantly to this time estimation. We use such computationally expensive pipeline to achieve the best localization results as the performance of visual localization network itself is not the aim of our study. We expect that this pipeline can be replaced by less demanding alternative, but this can lead to a localization quality drop.

9 Additional experiments and experimental setup

To additionally measure the human-object localization accuracy of our method, we recorded a special sequence which additionally has ground truth data obtained via an external depth multicamera system. The experiment setup almost



Ground truth from the multicamera system

Scene reconstruction by HOPS

Fig. 8: Visual comparison between the ground truth point cloud and our result. Left shows ground truth data from merging point clouds from 3 depth cameras, right shows a reconstruction of the same scene by HOPS as a combination of object, human model, and visible scan vertices.

exactly repeats the one from HPS [17], however here we capture the full dynamic object interaction while in HPS only the human body motion and the static scene was captured.

We use a system of 3 calibrated Azure Kinect [3] RGBD sensors – by combining the outputs of these sensors we obtain a sequence of 3D point clouds of the scene and a subject. Each sensor outputs the depth map with resolution of 640×576 pixels and color frames with resolution of 2048×1536 at around 30 FPS. Azure Kinect features build-in temporal synchronization, but to merge the sensors output into scene ground truth representation we also need to calibrate them spatially within the 3D scene coordinate system. For that we use a 3-stage localization pipeline, similar to [17]: **1)** we record a special sequence of the empty scene; for each frame of this 300-frame empty sequence we localize the RGB camera using the same visual localization algorithm used for the head-mounted camera, all 300 localizations are then averaged; **2)** we perform ICP between the scene 3D scan and the pointcloud unprojected from the depthmap; **3)** after that, we perform manual correction if needed. Using the obtained positions of the sensors, the pointcloud representation of the scene is formed by unprojecting depthmaps from all 3 sensors to 3D. To perform the evaluation, we manually synchronize the time between the HOPS motion sequence and the aforementioned pointcloud representation. For each frame of the test sequence we separately measure object and human localization accuracy E_{obj} and E_{body} :

- E_{obj} : mean Chamfer distance from the object point cloud to ground truth point cloud
- E_{body} : mean Chamfer distance from the human body SMPL mesh to ground truth point cloud

Results are presented in supp. Table 1. As seen from the table, HPS results in a much higher mean distance between the object representation and the ground truth point cloud because this method does not track scene changes. The metric improves when visual object localization is added before and after the interaction. However, the numbers do not relate entirely to the realism of the motion,

	HPS (Improved)	HPS (Improved) + Visual object localization	HOPS
Object to GT E_{obj}	23.346	8.557	8.330
SMPL to GT E_{body}	8.845	8.845	8.628

Table 1: **Quantitative evaluation:** 3D error (in cm) between the object and human models and ground truth point cloud captured by a synchronized Kinect recording setup. Results are shown for the method without dynamic scene adaptation, method with the visual object localization only and HOPS.

which can only be observed from the animated visualizations. *Therefore, the qualitative results are more important than the quantitative for the given problem.* As seen in the supplementary video, a simple interpolation between the starting and ending positions does not reflect the real motion. As a result, the human is constantly losing contact with the object. By contrast, the supplementary video shows greatly improved visual quality and realism of the motion with HOPS, even though the quantitative improvement is not as obvious. In addition to that, HOPS also contributes to better human body localization by correcting the pose based on the motion, meaning that body and object localization benefit from the joint tracking of both.

References

1. Project Aria (accessed March 1, 2022), <https://about.fb.com/realitylabs/projectaria/> 14
2. Autodesk FBX Software Developer Kit (accessed March 11, 2022), <https://www.autodesk.com/developer-network/platform-technologies/fbx-sdk-2020-0> 18
3. Microsoft Azure Kinect (accessed March 11, 2022), https://en.wikipedia.org/wiki/Azure_Kinect 19
4. Bhatnagar, B.L., Singh, S., Arora, C., Jawahar, C.: Unsupervised learning of deep feature representation for clustering egocentric actions. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 1447–1453 (2017). <https://doi.org/10.24963/ijcai.2017/200>, <https://doi.org/10.24963/ijcai.2017/200> 4
5. Brachmann, E., Humenberger, M., Rother, C., Sattler, T.: On the limits of pseudo ground truth in visual camera re-localisation. In: ICCV (2021) 5
6. Brachmann, E., Rother, C.: Learning Less is More - 6D Camera Localization via 3D Surface Regression. In: CVPR (2018) 5
7. Brachmann, E., Rother, C.: Visual camera re-localization from RGB and RGB-D images using DSAC. arXiv:2002.12324 (2020) 5
8. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000) 18
9. Cao, C., Zhang, Y., Wu, Y., Lu, H., Cheng, J.: Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. 2017 IEEE International Conference on Computer Vision (ICCV) (2017) 4
10. Cavallari, T., Golodetz, S., Lord, N.A., Valentin, J., Prisacariu, V.A., Di Stefano, L., Torr, P.H.S.: Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2019) 5
11. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 224–236 (2018) 15
12. Dong, S., Fan, Q., Wang, H., Shi, J., Yi, L., Funkhouser, T., Chen, B., Guibas, L.J.: Robust neural routing through space partitions for camera relocalization in dynamic indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8544–8554 (2021) 5
13. Doosti, B., Naha, S., Mirbagheri, M., Crandall, D.J.: Hope-net: A graph-based model for hand-object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6608–6617 (2020) 4
14. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 226–231. KDD'96, AAAI Press (1996) 9
15. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: 2011 international conference on computer vision. pp. 407–414. IEEE (2011) 4
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (Jun 1981). <https://doi.org/10.1145/358669.358692>, <https://doi.org/10.1145/358669.358692> 17

17. Guzun, V., Mir, A., Sattler, T., Pons-Moll, G.: Human positioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4318–4329 (2021) [2](#), [5](#), [7](#), [9](#), [13](#), [15](#), [19](#)
18. Hartley, R., Trunpf, J., Dai, Y., Li, H.: Rotation averaging. International journal of computer vision **103**(3), 267–305 (2013) [17](#)
19. Hassan, M., Choutas, V., Tzionas, D., Black, M.J.: Resolving 3D human pose ambiguities with 3D scene constraints. In: International Conference on Computer Vision. pp. 2282–2292 (Oct 2019), <https://prox.is.tue.mpg.de> [4](#)
20. Huang, Y., Kaufmann, M., Aksan, E., Black, M.J., Hilliges, O., Pons-Moll, G.: Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **37**(6), 185:1–185:15 (nov 2018) [14](#)
21. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From Structure-from-Motion Point Clouds to Fast Location Recognition. In: CVPR (2009) [5](#)
22. Jacquet, B., Angst, R., Pollefeys, M.: Articulated and restricted motion subspaces and their signatures. In: CVPR (2013) [3](#)
23. Jafarzadeh, A., Antequera, M.L., Gargallo, P., Kuang, Y., Toft, C., Kahl, F., Sattler, T.: Crowddriven: A new challenging dataset for outdoor visual localization. In: ICCV (2021) [5](#)
24. Jiang, H., Grauman, K.: Seeing invisible poses: Estimating 3d body pose from ego-centric video. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3501–3509. IEEE (2017) [5](#)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [18](#)
26. Kwon, T., Tekin, B., Stuhmer, J., Bogo, F., Pollefeys, M.: H2o: Two hands manipulating objects for first person interaction recognition. arXiv preprint arXiv:2104.11181 (2021) [4](#)
27. Labbé, Y., Carpentier, J., Aubry, M., Sivic, J.: Cosypose: Consistent multi-view multi-object 6d pose estimation. In: ECCV (2020) [6](#)
28. Li, Y., Snavely, N., Huttenlocher, D.P., Fua, P.: Worldwide Pose Estimation Using 3D Point Clouds. In: ECCV (2012) [5](#)
29. Liu, S., Jiang, H., Xu, J., Liu, S., Wang, X.: Semi-supervised 3d hand-object poses estimation with interactions in time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14687–14697 (2021) [4](#)
30. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Transactions on Graphics (2015) [8](#), [18](#)
31. Lynen, S., Zeisl, B., Aiger, D., Bosse, M., Hesch, J., Pollefeys, M., Siegwart, R., Sattler, T.: Large-scale, real-time visual-inertial localization revisited. The International Journal of Robotics Research **39**(9), 1061–1084 (2020) [5](#)
32. Ma, M., Fan, H., Kitani, K.M.: Going deeper into first-person activity recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1894–1903 (2016) [4](#)
33. Oberweger, M., Wohlhart, P., Lepetit, V.: Generalized feedback loop for joint hand-object pose estimation. IEEE transactions on pattern analysis and machine intelligence **42**(8), 1898–1912 (2019) [4](#)
34. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning

- library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 32, pp. 8024–8035. Curran Associates, Inc. (2019), <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf> 18
35. Paulich, M., Schepers, M., Rudigkeit, N., Bellusci, G.: Xsens MTw Awinda: Miniature Wireless Inertial-Magnetic Motion Tracker for Highly Accurate 3D Kinematic Applications (05 2018). <https://doi.org/10.13140/RG.2.2.23576.49929> 8, 15, 17
 36. Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.P., Schiele, B., Theobalt, C.: Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)* **35**(6), 162 (2016) 4
 37. Rogez, G., Supancic, J.S., Ramanan, D.: First-person pose recognition using egocentric workspaces. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4325–4333 (2015) 4
 38. Rünz, M., Agapito, L.: Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4471–4478. IEEE (2017) 2
 39. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 12716–12725 (2019) 5, 6, 8
 40. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperGlue: Learning Feature Matching with Graph Neural Networks. In: *CVPR (2020)* 6, 15
 41. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. *PAMI* (2017) 5
 42. Schönberger, J.L., Pollefeys, M., Geiger, A., Sattler, T.: Semantic Visual Localization. In: *CVPR (2018)* 5
 43. Shi, Y., Huang, J., Xu, X., Zhang, Y., Xu, K.: StablePose: Learning 6D Object Poses From Geometrically Stable Patches. In: *CVPR (2021)* 6
 44. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2013) 5
 45. Sofiiuk, K., Petrov, I., Konushin, A.: Reviving iterative training with mask guidance for interactive segmentation. *arXiv preprint arXiv:2102.06583* (2021) 8
 46. Su, Z., Xu, L., Zhong, D., Li, Z., Deng, F., Quan, S., Fang, L.: Robustfusion: Robust volumetric performance reconstruction under human-object interactions from monocular rgbd stream. *arXiv preprint arXiv:2104.14837* (2021) 2
 47. Sun, G., Chen, X., Chen, Y., Pang, A., Lin, P., Jiang, Y., Xu, L., Yu, J., Wang, J.: Neural free-viewpoint performance rendering under complex human-object interactions. In: *Proceedings of the 29th ACM International Conference on Multimedia*. pp. 4651–4660 (2021) 2
 48. Toft, C., Maddern, W., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Pajdla, T., Kahl, F., Sattler, T.: Long-Term Visual Localization Revisited. *TPAMI* pp. 1–1 (2020). <https://doi.org/10.1109/TPAMI.2020.3032010> 5
 49. Tome, D., Alldieck, T., Peluse, P., Pons-Moll, G., Agapito, L., Badino, H., de la Torre, F.: Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Oct 2020) 4
 50. Tsuru, M., Escande, A., Tanguy, A., Chappellet, K., Harad, K.: Online object searching by a humanoid robot in an unknown environment. *IEEE Robotics and Automation Letters* **6**(2), 2862–2869 (2021) 2

51. von Marcard, T., Rosenhahn, B., Black, M., Pons-Moll, G.: Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum* 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics) pp. 349–360 (2017) [14](#)
52. Wald, J., Avetisyan, A., Navab, N., Tombari, F., Niessner, M.: RIO: 3D Object Instance Re-Localization in Changing Indoor Environments. In: *ICCV* (2019) [8](#)
53. Wald, J., Sattler, T., Golodetz, S., Cavallari, T., Tombari, F.: Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In: *European Conference on Computer Vision*. pp. 467–487. Springer (2020) [5](#)
54. Wang, G., Manhardt, F., Tombari, F., Ji, X.: GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In: *CVPR* (2021) [6](#)
55. Weng, Z., Yeung, S.: Holistic 3d human and scene mesh estimation from single view images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 334–343 (2021) [4](#)
56. Wong, Y.S., Li, C., Niessner, M., Mitra, N.J.: Rigidfusion: Rgb-d scene reconstruction with rigidly-moving objects. *Computer Graphics Forum* 40(2) (2021) [4](#)
57. Xu, W., Chatterjee, A., Zollhoefer, M., Rhodin, H., Fua, P., Seidel, H.P., Theobalt, C.: Mo²Cap² : Real-time mobile 3d motion capture with a cap-mounted fish-eye camera. *IEEE Transactions on Visualization and Computer Graphics* pp. 1–1 (2019) [4](#)
58. Xu, X., Joo, H., Mori, G., Savva, M.: D3d-hoi: Dynamic 3d human-object interactions from videos. *arXiv preprint arXiv:2108.08420* (2021) [4](#)
59. Yang, Z., Yu, X., Yang, Y.: DSC-PoseNet: Learning 6DoF Object Pose Estimation via Dual-Scale Consistency. In: *CVPR* (2021) [6](#)
60. Yi, X., Zhou, Y., Xu, F.: Transpose: real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)* 40(4), 1–13 (2021) [14](#)
61. Yonemoto, H., Murasaki, K., Osawa, T., Sudo, K., Shimamura, J., Taniguchi, Y.: Egocentric articulated pose tracking for action recognition. In: *International Conference on Machine Vision Applications (MVA)* (2015) [4](#)
62. Yuan, Y., Kitani, K.: 3d ego-pose estimation via imitation learning. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 735–750 (2018) [5](#)
63. Yuan, Y., Kitani, K.: Ego-pose estimation and forecasting as real-time pd control. In: *The IEEE International Conference on Computer Vision (ICCV)* (October 2019) [5](#)
64. Zhang, J.Y., Pepose, S., Joo, H., Ramanan, D., Malik, J., Kanazawa, A.: Perceiving 3d human-object spatial arrangements from a single image in the wild. In: *European Conference on Computer Vision (ECCV)* (2020) [4](#)