DOI: 10.1111/cgf.14685
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

Joint Hand and Object Pose Estimation from a Single RGB Image using High-level 2D Constraints

H.-X. Song[1], T.-J. Mu[†1] and R. R. Martin[1,2]

[1]BNRist, Department of Computer Scienece and Technology, Tsinghua University, China
[2]Cardiff University, UK

Abstract

Joint pose estimation of human hands and objects from a single RGB image is an important topic for AR/VR, robot manipulation, etc. It is common practice to determine both poses directly from the image; some recent methods attempt to improve the initial poses using a variety of contact-based approaches. However, few methods take the real physical constraints conveyed by the image into consideration, leading to less realistic results than the initial estimates. To overcome this problem, we make use of a set of high-level 2D features which can be directly extracted from the image in a new pipeline which combines contact approaches and these constraints during optimization. Our pipeline achieves better results than direct regression or contact-based optimization: they are closer to the ground truth and provide high quality contact.

CCS Concepts

• ***Computing methodologies*** → ***Reconstruction;***

1. Introduction

Modeling and reconstructing the interactions between hands and objects, together with the localization and mapping technology [BK20, DHM*22, WJW*20, HYZ*21, HYMH20, XRW21] for the visual sensors, has inspired a wide range of applications in VR/AR [WLLZ20], robotic grasping, human-robot interaction, etc. Traditional methods such as [HVT*19, TBP19, KKB19, DNMC20, HTB*20] directly predict poses and states of objects and a parametric hand model from a single monocular image, using a unified neural network. Although such an approach can certainly ensure overall robustness, it is hard to recover a natural grasp, i.e. one in which there is appropriate contact between the hand and object, without intersections, yet with very little gap between their meshes, in the contact region. Therefore, many approaches have been applied to obtain a credible result, including distance-based attraction and repulsion [HVT*19, KYZ*20], learned contact regions [BHHF19, JLWW21, GTT*21], physical simulation [KKB19, KKB20, GHJK20]. Several recent works [GTT*21, YZL*21, CRKM21, ZZXW22] model the contact either by predicting contact [GTT*21, YZL*21, CRKM21] or analyzing physical forces [ZZXW22], and then apply optimization to refine an initial pose estimated by a traditional method.

However, existing methods usually separate optimization of initial poses from the high-level constraints conveyed by the input image, causing the final poses to deviate from realism. RHO [CRKM21] adds constraints provided by object masks and a depth map when estimating the poses, but fails to consider the hand during final refinement using a contact based approach. CPF [YZL*21] tackles the problem to a certain extent by directly relating image features to the object to predict a contact potential field. However, when the initial poses of hand and object disagree, a dilemma is faced: should we update the pose of the object or the pose of the hand? Without guidance from the original input image, if contact is the only criterion used in final optimization, errors in initial pose may be made worse, if the object or hand with more accurate pose is adjusted to bring it into better contact with the other.

To address the above challenges, in this paper, we propose a new optimization framework to estimate the hand-object (HO) pose from a single RGB image of hand and object interaction, by imposing useful constraints on the high-level features directly extracted from the input image. Together with the contact cues, our framework is capable of recovering a faithful hand-object pose in terms of model accuracy and contact quality. Specifically, to refine the initial hand-object pose to be close to the one implied in the input image, we first include the semantic segmentation masks of hand and object as a high-level 2D constraint. However, the semantic mask alone can only constrain the model with low degree of freedom, for the model with high degree of freedom and non-rigid motion like hands, more elaborate correspondences are required. We therefore introduce dense mapping features [WSH*19, LAZ*22] to

[†] Corresponding author. Email: taijiang@tsinghua.edu.cn

help to constrain the pose and shape. To further handle the depth ambiguities and occlusion inherent in a single view image, we also resort to the depth map. All these high-level features are directly extracted from the input image with a deep neural network.

Finally, these features are incorporated into the optimization framework with loss terms designed aware of the occlusion and error of the initial pose to constrain the optimized pose to lie close to the truth. On the other hand, we also apply contact terms in the optimization, ensuring the results both have high quality of contact, and are in agreement with the input image.

To evaluate our approach, we report reconstruction and physical quality on FHB [BTT*20a] and HO3D [HORL19, HROL20] datasets; our method achieves better results than the state-of-the-art methods for both direct regression and contact-based optimization: they are closer to the ground truth and provide high quality contact. We also provide an ablation study to show the effectiveness of each feature, and how it contributes to resolving depth ambiguities and occlusion, as well improving prediction errors.

In summary, the contributions of this paper are as follows.

- We are first to include dense image features for both hand and object when optimizing estimates of hand-object contact, ensuring adherence to the input image in addition to high quality contact.
- We provide a neural network to extract a set of high-level 2D features directly from the original image; they suffice to overcome inherent depth ambiguities and strong occlusion, allowing determination of accurate 3D information.
- We augment the optimization process with a set of terms based on the high-level 2D features, allowing it to better reduce errors in the initial pose estimates and contact prediction.

2. Related Work

2.1. Hand-Object Pose Estimation

There has been great progress in reconstructing or estimating the pose of a single hand [KS12, GRL*19, IMB*18, CCY*21, ZLM*19] or objects [HHFS19, KMT*17, PLH*19, ZSI19, LF20, ZHMW22, LZXQ21, YJLF22, CG22, ZBB21, SHCM21] alone over recent decades. Lacking good datasets labeling hands and objects together, early work on hand-object interaction focused on recovering either the hand [RKK09, RKI*14] or object [TG15] pose in a interaction. The emergence of large datasets of hand-object interactions [BTT*20a, CYX*21, HORL19, HROL20, BTT*20b, BHKH19, ZYSK21, TGBT20], allowed methods that simultaneously estimate both hand and object pose [HVT*19, HTB*20, TBP19, KKB19, DNMC20, KYZ*20, HVSL21, ZYSK21, HPSK21]. [HVT*19] gives a pioneering algorithm to reconstruct shapes and poses of both hand and object together by using additional synthetic data and contact terms, while [KYZ*20] recovers models using the signed distance function [PFS*19]. [ZYSK21] proposes an novel spatial representation to reconstruct manipulation motions of the fingers for a wide range of general object shapes. Other work [TBP19, KKB19, DNMC20, HTB*20, HVSL21] focuses on using known models for the hand state and object pose. We use one such representative work [HTB*20] to obtain an initial pose for the hand and object, which we then refine.

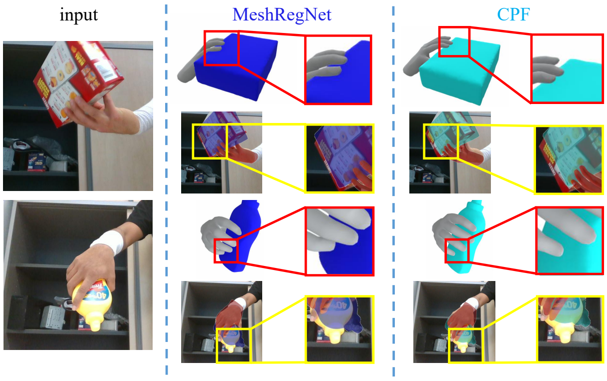

Figure 1: *Contact-based methods such as CPF [YZL*21] use an optimization step to overcome intersections and poor contact resulting from direct regression methods like MeshRegNet [HTB*20], but may not improve poor initial pose (below) or even worsen it while improving contact especially when the contact prediction is not so accurate (above).*

2.2. Contact-Based Methods

Contact heuristics can greatly improve the modeling of hand-object interaction. Physical simulation of contact in hand manipulations [YL12, KP06] can help avoid poses with intersecting models: [HOAL18, KP15] use the idea of contact points in the physical simulator to solve the penetration problem, while [CKA*22] uses contact terms to keep the grasping pose stable while generating hand motions to move an object. Many recent methods estimate the hand-object(HO) pose using some pre-defined contact terms. Some use a single term of attraction and repulsion between hand and object models [HVT*19, AGHK18, TBS*16, BHHF19, KYZ*20], while others use a predefined pattern [RIR15], but neither refer back to the original image. [NNN20]and [SGSF20] model 2D contact for the hand directly from the RGB image, while [GTT*21] and [CRKM21] learn 3D hand-object contact priors from well-labeled datasets [BTT*20b, TGBT20] using only 3D meshes as inputs. CPF [YZL*21] considers both 2D and 3D information. It uses a backbone to extract 2D latent features and projects them to the object's vertices in the directly regressed pose, then predicts the contact region on object vertices. It provides state-of-the-art results, but still is unable to improve initial pose estimates, and may even worsen them during optimization. This is the problem we address.

2.3. Semantic Features in Pose Estimation

Semantic features in images have long been used for estimating poses of objects and hands. Such features are usually used as intermediate clues in pose estimation. [BKK19] estimates hand pose from the 2D pose using extracted 2D features such as the of the hand mask and its 2D skeleton. [LAZ*22] adopts various dense features to supervise alignment of the hand mesh to the image. [ZZXW22] and [DNMC20] use 2D features to guide regression of final poses of both the hand and the object. While using 2D features, such papers just use them as clues during regression, instead of treating them as hard constraints during final optimization. As

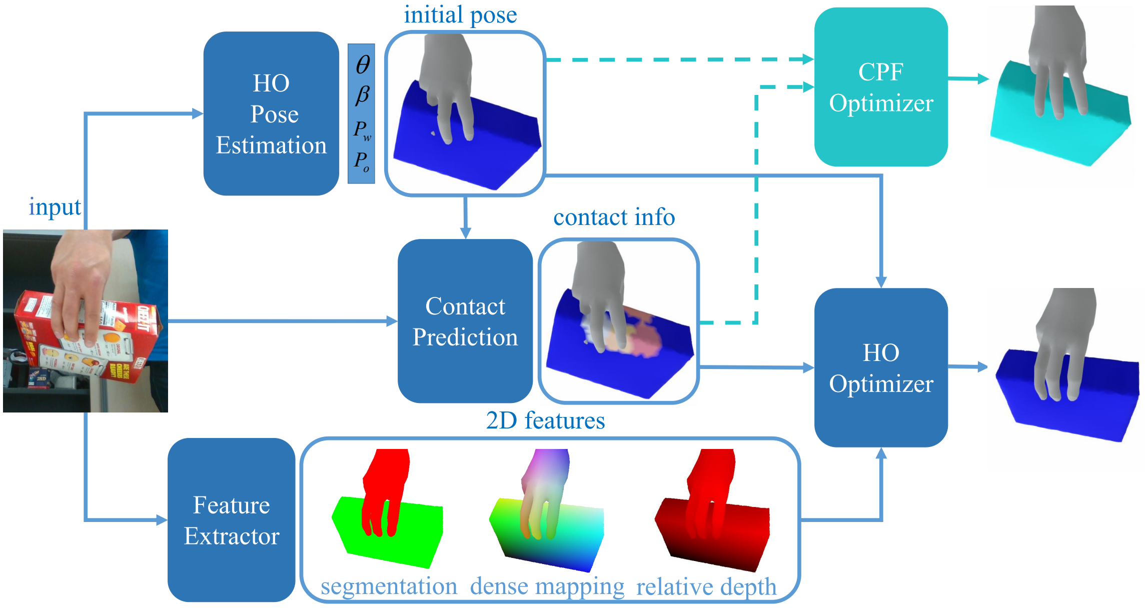

Figure 2: *Pipeline. Given a single RGB image as input, an initial pose is estimated by the HO Pose Estimation module using [HTB*20]. The meshes of hand and object, together with the RGB image, are used by the Contact Prediction module to obtain semantic contact information [YZL*21]. A Feature Extractor provides 2D features including semantic segmentation, dense mapping and relative depth maps directly from the original image. The HO Optimizer uses both 2D feature constraints and contact constraints to refine the hand-object pose. The dotted lines and CPF optimizer shows the pipeline of the baseline CPF [YZL*21], without consideration of the original 2D features.*

a result, these constraints are unlikely to be satisfied in the final result.

However, semantic features are used as constraints during the optimization phase in tracking and pose recovery. [WSH*19] estimates the poses of certain objects via predicting and fitting the coordinates of each pixel in normalized object coordinate space, while [HHFS19] estimates 6D object pose using a segmentation-driven method. [CLM*21] recovers the root position of the hand using adaptive 2D-1D registration of hand joints and the silhouette of the hand. [OKA12] recovers parameters of a hand model using skin color detection in RGB data together with a depth map. [MDB*19] reconstructs the poses and shapes of two interacting hands from a single depth image with the help of a semantic segmentation and a vertex-to-pixel map using the predicted dense correspondence. [WMB*20] tracks two hands without depth information by estimating and fitting the intra-hand and inter-hand depths of each pixel from the RGB image. Such research on either hands or objects inspires us to adopt dense semantic features as constraints in HO pose estimation. However, the strong occlusion and more complicated situations in HO interaction mean there is still much left to do.

3. Background

Our method is built upon CPF [YZL*21], which when refining the initial poses of hand and object considers only contact. To make our paper self-contained, we first briefly describe CPF.

In CPF, the hand is represented by A-MANO [YZL*21], a modified parametric skinning hand model derived from MANO [RTB17], which imposes restrictions on the joints' rotation axes and angles to avoid abnormal hand poses from arising during optimization. Specifically, a hand pose is described by an angle parameter $\theta = \{R_1, \cdots, R_{15}\}$ containing the 15 joint rotations $\{R_j \in SO(3) | 1 \leq j \leq 15\}$ in the hand kinematic tree, and a shape parameter $\beta \in R^{10}$ which represents principal component analysis (PCA) components of the hand shape. In addition, a 6D pose parameter $P_w \in SE(3)$ controls the pose of the root joint of the hand. The object, whose geometry is considered to be fixed and known, is simply described by a 6D pose $P_o \in SO(3)$.

As a typical contact-based optimization method, there are three parts to CPF [YZL*21] as shown in Figure 2, i.e., the HO Pose Estimation module, the Contact Prediction module and the CPF Optimizer. Firstly, a hand-object (HO) pose estimation network, MeshRegNet [HTB*20], is used to regress initial coarse poses of the hand and the object directly from the image. Then a contact prediction module determines contact between the hand and the object by projecting image features onto the vertices of the model using the initial poses. The contact information is of two kinds: the subregion of the hand each object vertex is likely to contact, and a parameter k^{atr} indicating the degree of attraction between the hand and the object, i.e., the closer the hand and object is, the larger k^{atr} is. Finally, a grasping optimizer (CPF Optimizer) is used to refine the initial HO pose taking into account the contact information and the hand's anatomical constraints. It does so by minimizing an objective function:

$$\mathcal{L}_{\text{opt}} = E_{\text{elast}} + \mathcal{L}_{\text{anat}} + \mathcal{L}_{\text{offset}}. \tag{1}$$

E_{elast} combines E^{atr} which encourages closeness of object vertices to various predicted subregions of the hand according to k^{atr}, and E^{rpl} which provides repulsive separation along the surface normal of hand and object vertices to prevent intersection. $\mathcal{L}_{\text{anat}}$ is used to avoid abnormal hand postures by constraining rotations of hand joints, and L_{offset} is a regularization term which penalizes change between the refined meshes and the initially predicted ones.

Although CPF [YZL*21] already takes the original image into consideration, it is still unable to guarantee consistency between the result and that image. Lacking direct constraints from the original image, two problems may arise. Firstly, optimization results will be close to the initial poses because of L_{offset}, so if the initial poses are too far out, the optimised result will still not agree with the image (see Figure 1(below)). Secondly, even if good initial poses are provided, a new problem will appear when ensuring contact: should we update the pose of the object or the hand? A wrong decision will lead to worse final poses (see Figure 1(above)). Additionally, the estimated contact information is imperfect. Without direct supervisory information from the image, there will be many solutions satisfying the region-based contact constraints, permitting further deviations from the true poses. To overcome these issues, we directly incorporate high-level features extracted from the input image into the contact-based optimizer to ensure HO poses having both good contact, and good agreement with the initial image.

4. Hand-Object Pose Estimation with 2D Constraints

4.1. Overview

To ensure that the predicted hand-object(HO) pose is in good agreement with the original input image, our new contact-based optimizer directly incorporates more information from the input image, in the form of a set of novel constraints. The full pipeline is illustrated in Figure 2.

Following CPF [YZL*21], given a single RGB image, we first use MeshRegNet [HTB*20] to directly regress the initial poses of the hand and the object, along with their contact positions. Meanwhile, a set of high-level 2D features of the image are also obtained by a feature extractor (see Sec. 4.3). These, together with the contact information, are input to our optimizer (see Sec. 4.2) as constraints to refine the initial HO pose.

4.2. HO Optimizer

4.2.1. Loss

Our optimizer iteratively refines the pose and shape of the hand and object pose jointly by adjusting the parameters of the models: P_o, P_w, $R_j$and β. The optimization process aims to minimize the loss function:

$$\mathcal{L}_{\text{opt}} = \mathcal{L}_{\text{contact}} + \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{offset}}, \tag{2}$$

where $\mathcal{L}_{contact}$ comes from the baseline method, CPF [YZL*21], and is given by $\mathcal{L}_{\text{contact}} = E_{\text{elast}} + L_{\text{anat}}$. $\mathcal{L}_{\text{offset}}$ also comes from CPF. In addition, we add a new loss $\mathcal{L}_{2D}$ which constrains the poses to agree with the high-level semantic features from the original image. We use three high-level features: semantic segmentation, dense mapping and relative depth, which are now described in detail.

4.2.2. Semantic Segmentation Term

A visible feature of the input image is the semantic segmentation of the hand and object, allocating each pixel to one or the other (or neither). The projected silhouettes of the posed hand and object should agree with boundaries in the semantic segmentation. This feature is defined using two class probability maps $\mathcal{S}_h, \mathcal{S}_o \in [0,1]^{h\times w\times 1}$ for hand and object respectively; w and h are the width and height of the image.

To disambiguate possible overlaps between the hand and object masks, we introduce an occlusion-awareness term when using of the feature. Inspired by neural rendering, we use N3MR [KUH18] which renders 3D models into an image, and shows how an approximate gradient for rasterization can allow rendering to be used in a gradient-based optimization framework. Because of the strong occlusion between hand and object, however, we cannot directly adopt the original silhouette term in [KUH18]. Firstly, since the neural renderer mostly affects those points with smallest depth, we should render the mask of object and hand separately to avoid vanishing gradients for those occluded parts which are thought to be visible because of incorrect initial pose. However, rendering them separately leads to a new problem. The neural renderer renders the mask for the whole model while the segmentation mask predicted from the original image only represents the visible part of the model. This will draw vertices of the model to the occlusion boundary and shrink the hand or push the object far away. Setting both predicted and rendered masks to zero and using their mean intersection over union (mIoU) as loss is also unworkable, since the gradients of visible pixels correspond to pixels of the occluded part, again keeping the model away from the occluded part. Thus, we adopt mean square error (MSE) between pixels from the predicted masks and rendered ones, while setting the distance of pixels belonging to the other mask to zero. Formally, this loss function can be expressed as:

$$\begin{aligned}\mathcal{L}_{\text{seg}} = \,&\|(1-\mathcal{S}_h)(\Pi_{\mathcal{S}}(\mathcal{V}_o,\mathcal{F}_o)-\mathcal{S}_o)\|^2 + \\ &\|(1-\mathcal{S}_o)(\Pi_{\mathcal{S}}(\mathcal{V}_h,\mathcal{F}_h)-\mathcal{S}_h)\|^2\end{aligned} \tag{3}$$

where $\Pi_{\mathcal{S}}(\cdot)$ denotes neural rendering of the segmentation mask, and $\mathcal{V}$ and $\mathcal{F}$ represent the vertices and faces of the models.

4.2.3. Dense Mapping Term

Although the semantic segmentation term can help to restrict solutions for the hand and object, there are still many possible local solutions for the HO pose, especially for the hand which has many degrees of freedom. Therefore, it is important to obtain a detailed vertex-level semantic feature as a dense mapping to further restrict the solution. To establish a dense semantic connection from the image to the model, we embed the models vertices $\mathcal{V}$ ($\mathcal{V}_o$ for the object and $\mathcal{V}_h$ for the hand) into a k-dimensional continuous space as $\varphi : \mathcal{V} \to R^k$ and define the dense mapping feature as a 2D map with k channels, where the value is determined by the φ value of the point projected to it. To embed different objects into

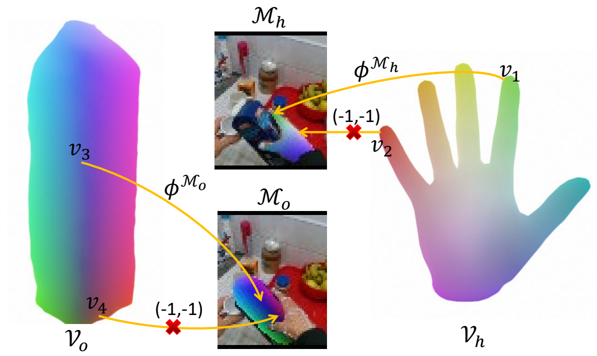

Figure 3: *Dense mapping of an object (left) and a hand (right). Each vertex $v \in \mathcal{V}_o(\mathcal{V}_h)$ of the object (hand) is mapped to a unique 3d feature, such that a dense correspondence $\phi^{\mathcal{M}}$ from the 3D vertex v to 2D pixel can be established by finding the nearest neighbour of the vertex $\mathcal{N}^{\mathcal{M}}(v)$ in the predicted dense mapping feature $\mathcal{M} \in \{\mathcal{M}_o, \mathcal{M}_h\}$. For the vertices which can not find a close enough corresponding pixel, we map them to (-1,-1) and discard them in the final refinement.*

the same coordinate space, we use the idea of normalized coordinate space [WSH*19]: we resize the object to fit into a unit cube, and use the coordinates $x,y,z \in [0,1]$ as the features of the vertices. For the hand, with more degrees of freedom, normalized coordinates are not enough. To distinguish different parts of the MANO model, we follow [LAZ*22] to get a positional embedding for each vertex. The projection assigns different colors for different vertices, especially in the regions of fingers, the most sensitive part when fitting the hand model. A visualization of the two embedding methods for the object and hand is shown in Figure 3. Since both the normalized coordinate space and the positional embedding of the hand are three-dimensional, we finally define the feature as a pair of three-dimensional maps $\mathcal{M}_h, \mathcal{M}_o \in [0,1]^{h\times w\times 3}$ for the object and hand respectively.

To make use of the dense mapping feature $\mathcal{M} \in \{\mathcal{M}_o, \mathcal{M}_h\}$, we first establish a direct correspondence from model vertices ($\mathcal{V} = \mathcal{V}_o$ for $\mathcal{M}_o$, $\mathcal{V} = \mathcal{V}_h$ for $\mathcal{M}_h$) to the RGB image by searching for the nearest pixel to each vertex in the feature space φ according to $\mathcal{M}$:

$$\mathcal{N}^{\mathcal{M}}(v) = \arg\min_{i\in\mathcal{I}} \|\mathcal{M}(i) - \varphi(v)\|_2 \tag{4}$$

Then we define $\phi^{\mathcal{M}} : \mathcal{V} \to \mathcal{I}$, which represents the corresponding pixel $i \in \mathcal{I}$ in the input image for each vertex v obtained by the certain dense mapping feature. For the vertices which can not find a close enough corresponding pixel, we map them to $(-1,-1)$ as outliers.

$$\phi^{\mathcal{M}}(v) = \begin{cases} \mathcal{N}^{\mathcal{M}}(v), & \text{if } \|\mathcal{M}(\mathcal{N}^{\mathcal{M}}(v)) - \varphi(v)\|_2 < \text{ threshold,} \\ (-1,-1), & \text{otherwise} \end{cases} \tag{5}$$

After finding the correspondences, for each available pair, we penalize the distance between the corresponding pixel and the position of the vertex projected by the camera. The loss function is:

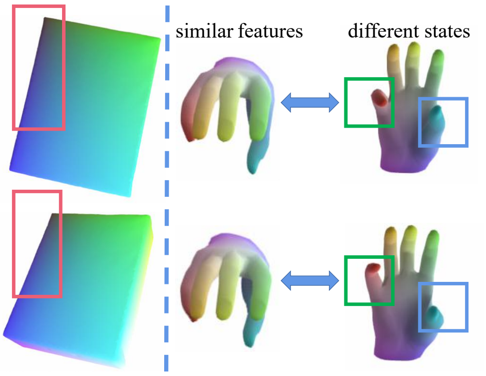

Figure 4: *Ambiguities. Alternative states can have similar segmentation maps and dense mapping in a certain point of view. Blue and green boxes point out the visible differences in the states of hand figures, while the red boxes point out the only big difference in poses of an object, but if occluded, this is hard to recognize.*

$$\mathcal{L}_{\text{dense}}^{\mathcal{M},\mathcal{V}} = \sum_{v\in\mathcal{V}\&\phi^{\mathcal{M}}(v)\neq(-1,-1)} \|\pi(v) - \phi^{\mathcal{M}}(v)\|^2, \tag{6}$$

where $\pi(\cdot)$ refers to the perspective camera projection from the camera space to the 2D plane of the image. Finally, the total dense mapping loss function for both the object and hand is:

$$\mathcal{L}_{\text{dense}} = \mathcal{L}_{\text{dense}}^{\mathcal{M}_o,\mathcal{V}_o} + \mathcal{L}_{\text{dense}}^{\mathcal{M}_h,\mathcal{V}_h}. \tag{7}$$

Although 2D joints of the hand could provide an alternative semi-dense feature to constrain hand pose, we choose not to adopt them in our optimizer. On one hand, the visible parts of joints are already embedded in the dense mapping. On the other hand, although many existing works predict the joints well for the hands, it is still an ill-posed problem to directly predict the invisible parts, especially when the hand is strongly occluded, in hand-object interaction. This can be seen from the failure in the 2nd row of Figure 6. We believe that semantics-based contact can give better constraints on the invisible part of the hand.

4.2.4. Relative Depth Term

2D images lack depth information, and it is important to eliminate the resulting ambiguity: see Figure 4. For the hand, there exist many reasonable states (pose and shape) corresponding to views of similar semantic segmentation and dense mapping features, because of the large number of degrees of freedom. For objects, although only rigid transformations can be applied, if the most discriminative part is occluded, it is still difficult to recognize the poses with only similar visible features. On the other hand, the depth map of object and hand can vary a lot in such situations, making it possible to constrain the pose refinement.

However, it is difficult to predict absolute depths relative to the camera accurately because of the bas-relief ambiguity [BKY97] as well as the large variation of depth among different scenes. Instead, we make use of relative depth, which is the distance of the model to its root point along the camera direction. For an object, we consider the centre of its bounding box as its root point. For the hand, instead of estimating the depths of the pixels to the root joint of the hand as in [WMB*20], we still compute the distance to the object center as the relative depth of the hand, because the large occlusion of the hand would make it difficult to determine the position of the hand root from the image, while the shape of the object is already known in the setting of the task.Therefore, we define the relative depth of the object and hand as a single map $\mathcal{D} \in [-1,1]^{h\times w\times 1}$.

Similar to the semantic segmentation term, a neural renderer is also used to render the depth. Although the relative depths of the hand and the object have the same root, we should render their depth separately. This is because, with a bad initial pose estimate, a pixel supposed to be visible as part of the hand may be belong to the object or vice versa, resulting in incorrect conclusions when raycasting. In addition, some parts of the mesh would be projected to the pixel where supposed to be the background, which will also relate the parts to the wrong depth, so we only consider pixels belonging to each corresponding segmentation. The loss function can be expressed as:

$$\begin{aligned}\mathcal{L}_{\text{dep}} = \|\mathcal{S}_o(\Pi_{\mathcal{D}}(\mathcal{V}_o,\mathcal{F}_o) - z_{root} - \mathcal{D})\|^2 + \\ \|\mathcal{S}_h(\Pi_{\mathcal{D}}(\mathcal{V}_h,\mathcal{F}_h) - z_{root} - \mathcal{D})\|^2,\end{aligned} \tag{8}$$

where $\Pi_{\mathcal{D}}(\cdot)$ means the process of neural rendering of the depth. z_{root} means the z-coord of the root point of the object.

However, the gradient of the $\mathcal{L}_{\text{dep}}$ cannot be transferred to the correct corresponding vertex because of the self-occlusion caused by the wrong initial pose. To deal with the situation, we also combine the knowledge of relative depth with dense mapping to define a new depth loss $\mathcal{L}_{\text{dep2}}$. For each vertex that has a correspondence in the image, its depth should be also close to the predicted depth of the pixel. Therefore, we can define the term as:

$$\mathcal{L}_{\text{dep2}}^{\mathcal{M},\mathcal{V}} = \sum_{v\in\mathcal{V}\&\phi^{\mathcal{M}}(v)\neq(-1,-1)} \left\|z_v - (\mathcal{D}(\phi^{\mathcal{M}}(v)) - z_{root})\right\|^2, \tag{9}$$

$$\mathcal{L}_{\text{dep2}} = \mathcal{L}_{\text{dep2}}^{\mathcal{M}_o,\mathcal{V}_o} + \mathcal{L}_{\text{dep2}}^{\mathcal{M}_h,\mathcal{V}_h}. \tag{10}$$

where z_v means the z-coord of the vertices of the models.

By making a direct correspondence between depth and the vertex, it is more direct to adopt the constraints of the relative depth free from incorrect mapping caused by the wrong initial pose. However, considering that $\mathcal{L}_{\text{dep2}}$ relies on two features which are predicted, there may be a notable deviation to the ground truth, so we still keep the previous term $\mathcal{L}_{\text{dep}}$.

Therefore, the final 2D constraint loss is computed as follows:

$$\mathcal{L}_{2\text{D}} = \lambda_{\text{seg}}\mathcal{L}_{\text{seg}} + \lambda_{\text{dense}}\mathcal{L}_{\text{dense}} + \lambda_{\text{dep}}\mathcal{L}_{\text{dep}} + \lambda_{\text{dep2}}\mathcal{L}_{\text{dep2}}. \tag{11}$$

where λ_{seg}, λ_{dense}, λ_{dep}, and λ_{dep2} are weights balancing the important of each term.

The whole optimization can be expressed as:

$$P_o^*, P_w^*, \beta^*, R_j^* = \underset{P_o,P_w,\beta,R_j}{\arg\min}\ \mathcal{L}_{\text{contact}} + \mathcal{L}_{2\text{D}} + \mathcal{L}_{\text{offset}}. \tag{12}$$

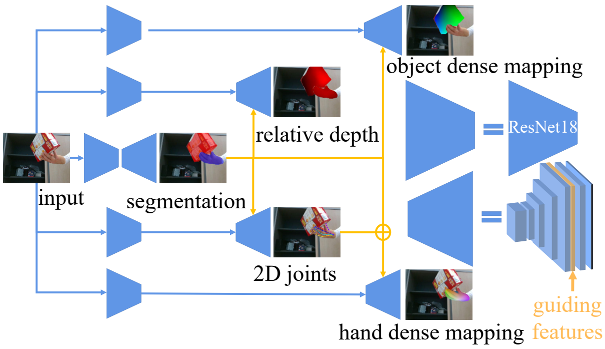

Figure 5: *Network architecture of the feature extractor. For each feature, we use ResNet18 to encode the input image, whose feature is decoded together with the guiding features to obtain the final feature map.*

Although the same loss terms of 2D feature above could be used to train the initial pose estimator as well, We decide to adopt them in the optimization step. Firstly, the above optimization can directly optimizes the pose parameters of hand and object to match the input given only a single image, while using them as training loss terms would try to optimize the parameters of the neural network to fit the whole dataset to keep the constraints satisfied, which is much harder to be accurate. Furthermore, it is known that the contact term performs better with an optimization step [YZL*21]. To make full use of the contact prior, we resort to the optimization above, and the 2D features are required as global constraints in the optimization step to draw the pose close to the input image. Therefore, we adopt the terms in HO Optimizer instead of initial pose estimator.

4.3. Feature Extractor

4.3.1. Training Data Generation

To train the feature extractor, we generate ground truth feature maps from ground truth hand and object information. Given parameters and meshes of the hand and object, we project the mesh to generate segmentation feature maps and dense mapping using the known camera parameters. While the datasets we use for training and evaluation have ground truth depth collected by sensors, it is noisy and contains holes. Furthermore, although the supplied poses may not be accurate, we cannot avoid using them to compute the depth of the root point. Therefore, we also use depths generated from the poses.

Also, although we do not use hand joints as a feature, they are still useful to guide generation of dense mapping of the hand. Therefore, we also predict a 21 channel heatmap for the 21 joints of the hand. Each channels is a Gaussian distribution centered at the 2D keypoint position with a radius of 3 pixels, and unit amplitude.

4.3.2. Network Architecture

In practical, for an acceleration of the optimization step, we extract and adopt the feature maps of width and height 1/4 of the original high-resolution image. We use ResNet [HZRS16]-based

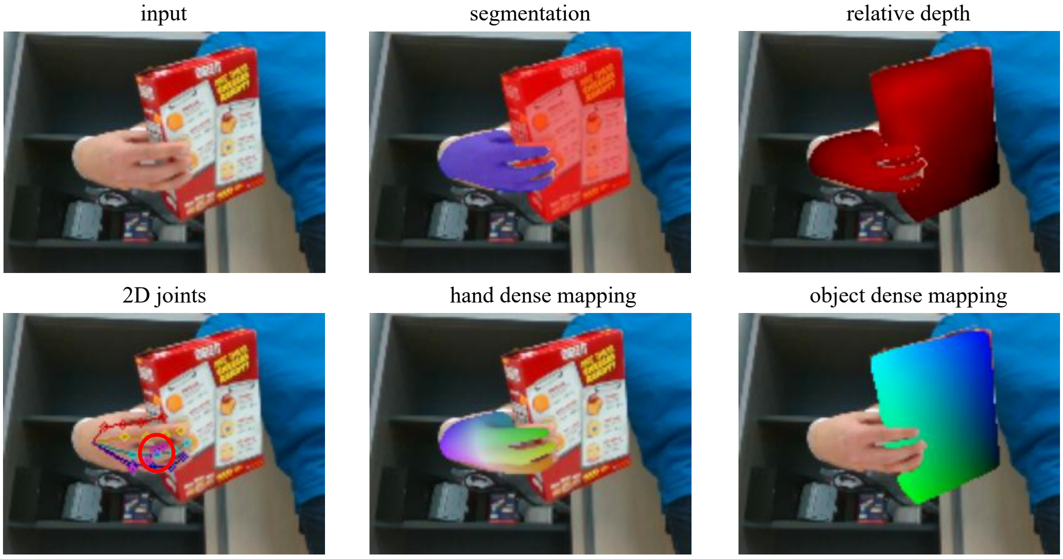

Figure 6: *An example of 2D features extracted by our feature extractor. The predicted joints of the little finger (purple ones) in the 2nd row are somewhat incorrect because of occlusion, while the part is not embedded by the dense mapping; this justifies our decision not to use 2D joints as a feature.*

hierarchical network to extract the features as shown in Figure 5. Specially, we use ResNet18 as an encoder to obtain the image feature. In each decoder, the feature encoded by ResNet with the size of $h/32 \times w/32 \times 512$ first goes through a convolutional layer reducing feature dimensions from 512 to 256, then 3 layers of upsampling with convolution to obtain a feature map of size of $h/4 \times w/4 \times 256$. Finally, the feature map is concatenated with the guiding features and passed through two convolutional layers to get the final feature map of width and height 1/4 of the original image. Additionally, the segmentation, joint heatmap and dense mapping feature pass through a sigmoid layer to limit values to (0, 1), while the relative depth map goes through a tanh layer to limit values to (-1, 1).

5. Experiments

5.1. Datasets

We evaluated our method on the real-world datasets FHB [BTT*20a] and HO3D [HORL19, HROL20] using the same settings as for the baseline method CPF [YZL*21].

FHB is a first-person RGBD video dataset of a hand manipulating objects. Following [YZL*21], we adopted a subset of FHB containing sequences of 4 objects, and the *action* split defined by [HTB*20]. We filtered the datasets to keep cases whose Euclidean distance between the hand and object is less than 5 mm, to ensure there is interaction between hand and object, resulting in 7223 samples for training and 7373 samples for testing. Note that the ground truth hand-object pose of FHB dataset is poor with many interpenetration due to the error of multi-view pose estimation.

HO3D is a hand-object interaction dataset with precise pose labeling. Two versions of HO3D are used in CPF, v1 and v2. Again following [YZL*21], we removed samples over a 5 mm distance threshold. For HO3Dv1, since the augmented data described in [YZL*21] is not provided, we just compare our methods with baselines on real data. For HO3Dv2, we select the same samples as test set HO3Dv2$^-$ in [YZL*21].

Table 1: *The weighting parameters of the loss terms for different datasets.*

Dataset	λ_{seg}	λ_{dense}	λ_{dep}	λ_{dep2}
FHB	1×10^{-2}	1×10^{-3}	1×10^{-2}	5×10^{-4}
HO3Dv1	5×10^{-2}	5×10^{-2}	5×10^{-1}	1×10^{-3}
HO3Dv2$^-$	5×10^{-3}	1×10^{-3}	5×10^{-3}	5×10^{-4}

5.2. Implement Details

Training of the Feature Extractor. To ensure the network can learn the right information from the guiding feature, we first train the segmentation sub-network to convergence, then we fix its parameters and train the relative depth and joint heatmap sub-networks to convergence, next, we fix their parameters and train the two dense mapping sub-networks, and finally, we train all parameters for another 20 epochs. We use cross-entropy loss for prediction of semantic segmentation and L1 loss for all other features. We employ an SGD optimizer with a weight decay of 1×10^{-4}. The learning rate is set to be varying linearly between 1×10^{-2} and 1×10^{-4} with a cycle of 150 steps. In addition, we perform data augmentation including rotation and Gaussian blurring of input images to increase the diversity of the training data. We implement our network with a highly efficient deep learning framework,

Table 2: *Quantitative results and detailed comparisons with previous methods for both direct pose regression and contact-based optimization on FHB and HO3D datasets. "gt." denotes the ground truth.(* Our SD values do not match those in the original paper [YZL*21], which we believe is due to differences in simulation settings.)*

Dataset	FHB				HO3Dv1				HO3Dv2$^-$		
Method	Ours	gt.	CPF	MeshRegNet	Ours	gt.	CPF	MeshRegNet	Ours	CPF	MeshRegNet
HE (mm)↓	18.08	0	19.54	**17.51**	**24.15**	0	27.20	26.85	**35.08**	35.69	37.28
OE (mm)↓	**20.12**	0	21.57	21.06	**19.26**	0	28.39	26.44	**65.40**	69.03	69.24
PD (mm)↓	18.19	19.55	**16.92**	20.63	12.43	**7.55**	9.61	19.91	**15.73**	16.47	20.02
SIV (cm^3)↓	14.45	20.41	**11.76**	21.10	3.49	3.57	**3.05**	10.71	**6.28**	7.44	9.25
SD* (mm)↓	67.42	67.76	**66.23**	68.70	**32.13**	18.46	53.48	33.25	**47.65**	50.92	52.69

Jittor [HLY*20], Figure 6 shows 2D features extracted from RGB images.

Optimization. For each sample, we optimize the HO pose by minimizing the loss function in 300 iterations, with an initial learning rate of 1×10^{-2}, reduced on plateau that the loss function has stopped decaying in 20 consecutive iterations. We employ a regular Adam solver when updating the arguments. The specific weights of terms in $\mathcal{L}_{2d}$ for different datasets are shown in Table 1. In addition, the threshold for filtering the invalid dense mapping pairs in Equ. 5 is set to be 0.05 for FHB and HO3Dv1, and 0.1 for HO3Dv2$^-$. These hyper-parameters are empirically set, mainly according to the baseline [YZL*21] (the contact term weight is larger for FHB, so is ours). For high quality datasets such as HO3D, the feature extractor can learn better features, so we increase the weights of the depth and dense mapping terms. If the initial pose is poor (e.g. as in HO3Dv2-), our terms will be much larger, so we decrease the weights of our terms to balance the contact term.

5.3. Metrics

To evaluate correspondence of results to the ground truth, we compute the mean per vertex position error (MPVPE) between the predicted mesh and the ground truth for both the hand (HE) and object (OE). We refer to this as state error.

We also evaluate the quality of contact using three metrics. The maximum penetration depth (PD) and the solid intersection volume (SIV) [HVT*19] are used to evaluate how deep the hand is penetrating the object's surface. In addition, simulation displacement (SD) is used to evaluate the contact stability. Our simulation is performed following [HVT*19] and its settings, in a modern physical simulator[†]. We do not adopt the disjointedness distance used in [YZL*21], because it is not always true that all five fingers should be close to the object surface (see the last example in Figure 8). For all the metrics, the smaller is better.

5.4. Comparison to Baselines

We compared our method with two state-of-the-art methods: MeshRegNet [HTB*20] as a baseline for directly regressing hand-object pose, and CPF [YZL*21] as a baseline for refining hand-object pose using contact-based optimization. A quantitative comparison on FHB, HO3Dv1 and HO3Dv2$^-$ datasets is shown in Table 2. For FHB [BTT*20a], we obtain the smallest Object MPVPE and have a Hand MPVPE just 0.57 mm larger than MeshRegNet; the latter provides much worse contact quality. Following the discussion in [BTT*20b], the ground truth hand-object pose in FHB is poor with many interpenetration due to the error of multi-view pose estimation, so it is hard to keep all the metrics low. Since we get a smaller sum of MPVPE and much better contact quality than MeshRegNet, and get 3 mm smaller MPVPE than CPF while only getting 1.3 mm larger in maximum penetration, we can say that our method is comparable to these state of the art methods.

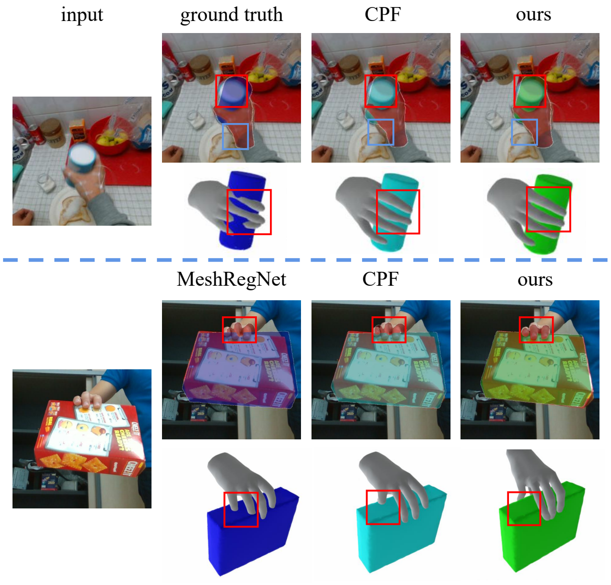

Figure 7: *Qualitative comparison with ground-truth and other methods on FHB (top) and HO3D (bottom) datasets. Our method provides an HO pose closer to the correct result, as verified by the mask projected onto the original image. Our results also provide good contact quality. In addition, we can see the ground truth of FHB is not so correct, which is one of the reason we don't get much improvement in the contact quality at the same time.*

For Ho3dv1, since we do not have access to the augmented data described in [YZL*21], we just compare our methods with the baselines on the real data. We get much better results in both the

[†] https://pybullet.org

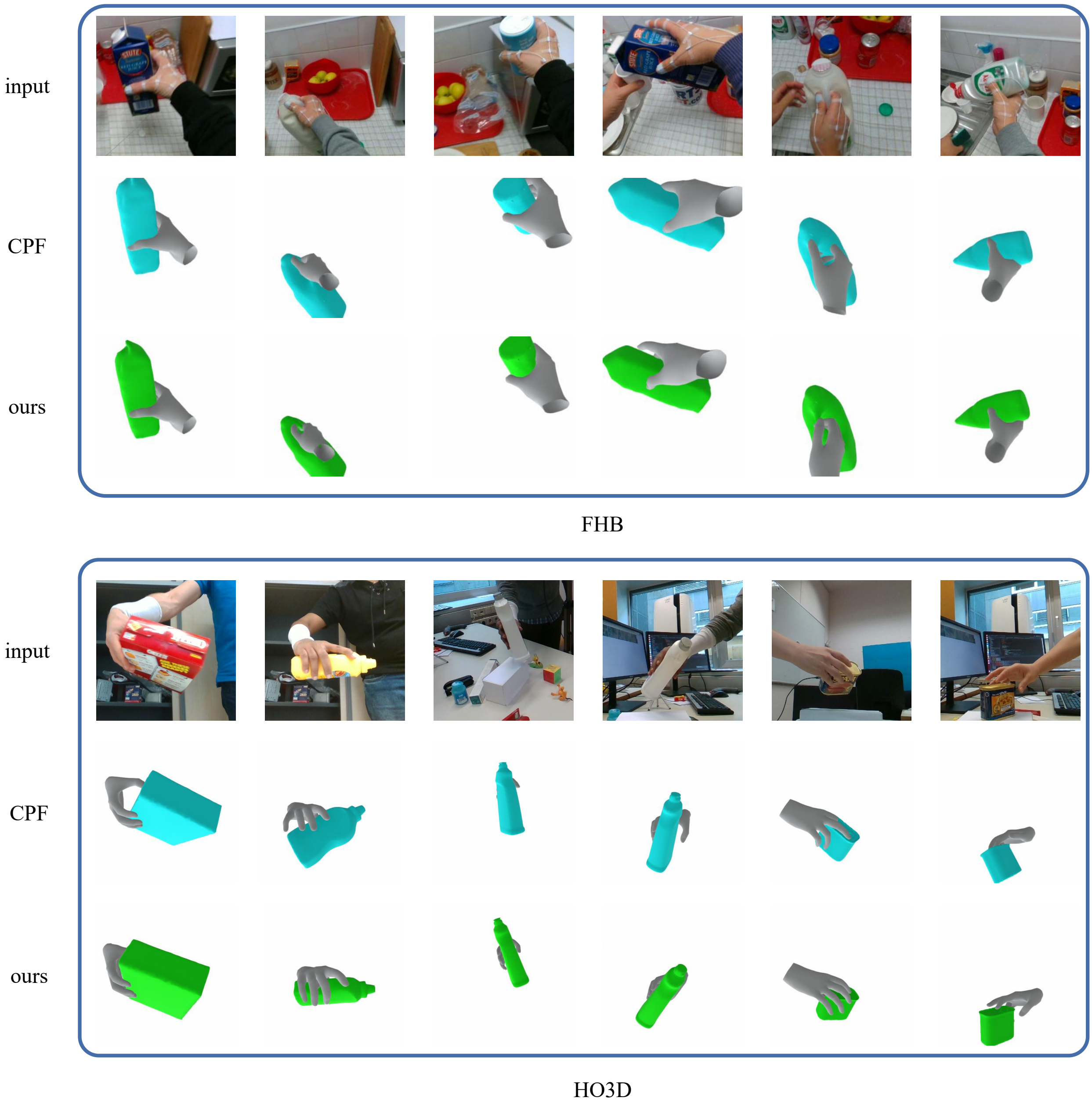

Figure 8: *Further results on FHB and HO3D datasets. The reconstructed hand and object are shown in the same view of the original image.*

Hand MPVPE and Object MPVPE, although we get about 2.8 mm larger in the max penetration, the displacement is much smaller, as Figure 1(top) shows, we argue that the CPF [YZL*21] may have a little excessive pursuit of the repulsive effect between the hand and object which separates them too much and results in a low intersection. In addition, since the ground truth has the smallest displacement and penetration depth, we can say the ground truth has a good contact, and a lower Hand MPVPE and Object MPVPE may indicate a better contact quality.

For Ho3dv2^{-}, unlike previous experiments, we try to keep the object in the real place as in the input image, so we should judge the quality of reconstruction by global position. Since there is no public

Table 3: *Results for different feature settings on the HO3Dv1 dataset. S: segmentation, M: dense mapping, D: relative depth.*

Settings			HE↓	OE↓	PD↓	SIV↓	SD↓
$\mathcal{S}$	$\mathcal{M}$	$\mathcal{D}$	(mm)	(mm)	(mm)	(cm^3)	(mm)
✓			25.58	23.42	10.28	2.75	47.82
	✓		25.42	34.97	8.56	1.01	51.56
		✓	26.80	33.21	11.29	2.61	37.27
✓	✓		25.41	22.34	9.59	2.38	45.01
✓		✓	25.24	31.50	13.59	4.55	34.11
	✓	✓	25.18	33.71	11.43	3.06	39.47
✓	✓	✓	**24.15**	**19.26**	12.43	3.49	**32.13**

ground truth of the hand except the position of the root joint, we just define the Hand MPVPE as the distance between the root joint of the predicted hand and the ground truth. Our method outperforms other methods in terms of all the metrics. One reason for the better results than on the other two datasets is the test set of Ho3dv2^{-} contains more difficult examples. Therefore, the predicted initial poses are much worse (see the HE and OE metrics in Table 2) and our method is more capable of drawing initial poses to the right ones with the help of 2D feature constraints.

Figure 7 shows qualitative results of our methods compared to baselines and the ground truth.

Our results are closer to the input image as well as providing good contact. It can also be seen that the ground truth of the FHB dataset indeed suffers from severe interpenetration, and the pose is sometimes rather incorrect, confirming our explanation above. More qualitative results can be seen in Figure 8.

While there is a physical-based approach, SCR [ZZXW22], which uses quite a different optimization process to our method. However, since its improved results are partly due to the design of its initial pose regression network, which lacks a detailed description. Differences in definition of the hand in the optimization step further make comparisons with this method difficult. We believe that, if adapted to the initial pose regression network and physics-based constraints of SCR [ZZXW22], our method would achieve better results.

5.5. Ablation Study

In this experiment, we evaluated our algorithm using different combinations of features on the HO3Dv1 dataset to demonstrate the efficacy of each feature. As Table 3 shows, each feature makes a contribution to pose constraints. Although the trend of the terms of contact quality are not so convincing, lower PD and SIV values do not always imply better and more correct results; instead, they only measure the intersection of hand and object. In most settings of the ablated experiments, the object is further from the correct position (see the OE values, indicating the distance between the ground truth and the predicted mesh) and the hand and object are far from each other. In these cases, the values of PD and SIV are 0, but the results are incorrect. Since the ground truth for HO3Dv1 is accurate, a smaller HE, OE and SD is more convincing, the value of SD will be much larger if the hand is not in contact with the object as shown in the final columns of Table 3.

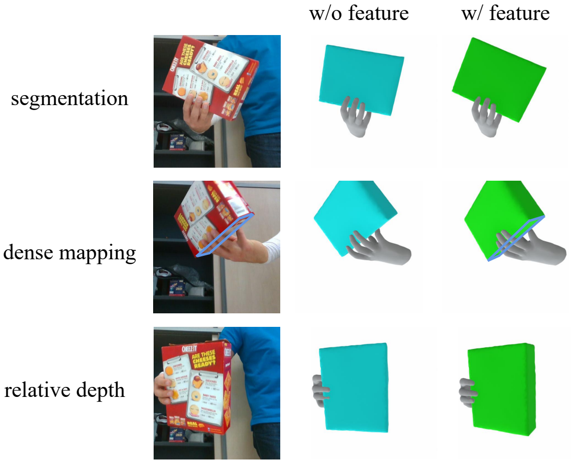

Figure 9: *Improvements provided by the three new features.*

In addition, we show several examples with and without the constraints of certain feature in Figure 9 to illustrate the efficacy of each feature. As we can see in the first row, there is an obvious improvement in the pose of the object when using the semantic segmentation constraint, which helps to adjust the object to the position indicated in the image with an accurate constraint to the silhouette of the object.

For the dense mapping, we can see two improvements in the example: First, the middle three fingers get closer to each other as in the input image with the dense mapping feature, which is because the mask and depth can only provide a general scope of the whole hand while the dense mapping feature provides more detailed correspondences to help to find more correct pose and shape for the hand. Second, the pose of the box is also improved (see the bottom of the box marked with blue lines). As we discussed in Sec. 4.2, with a bad initial pose (as the result w/o feature shows), the bottom of the box will be self-occluded by the object and cannot be corrected by the neural renderer, while the dense mapping can still establish a direct correspondence and pull the box to the right pose.

For the relative depth, we can see obvious improvements in both the pose of the hand and the object. In this case, the constraint of the relative depth map is necessary since the segmentation and dense mapping features are very similar in the given point of view, making it impossible to overcome the depth ambiguities only with these two terms.

6. Conclusion

This paper proposes a new, contact-based, simultaneous hand and object pose optimization method which directly incorporates high-level 2D features extracted from a single RGB image. Both quantitative and qualitative evaluations show that our method can recover more realistic hand-object pose while ensuring good contact quality. We hope that our work can provide new considerations in simultaneous hand-object reconstruction and inspire work that considers both agreement with the input image, and physical constraints. In the future, we plan to combine our method with other contact information such as physical forces [ZZXW22], and extend the constraints to an object-agnostic version for more general situations [HVT*19]. In addition, incorporating the proposed 2D constraints into the training of the regression of hand and object pose to enable a more effective method for end-to-end hand-object pose estimation is also a valuable direction. We would also like to consider better metrics to take both the quality of model geometry and the interpenetration between the hand and object into account.

Acknowledgements

We thank all the anonymous reviewers for their valuable suggestions. This work is supported by the National Natural Science Foundation of China (Grant No.: 61902210).

References

[AGHK18] Antotsiou D., Garcia-Hernando G., Kim T.-K.: Task-oriented hand motion retargeting for dexterous manipulation imitation. In *ECCV* (2018), vol. 11134, pp. 287–301. 2

[BHHF19] Brahmbhatt S., Handa A., Hays J., Fox D.: Contactgrasp: Functional multi-finger grasp synthesis from contact. In *IEEE/RSJ IROS* (2019), pp. 2386–2393. 1, 2

[BHKH19] Brahmbhatt S., Ham C., Kemp C. C., Hays J.: Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *IEEE/CVF CVPR* (2019), pp. 8709–8719. 2

[BK20] Bodonyi A., Kunkli R.: Efficient object location determination and error analysis based on barycentric coordinates. *Visual Computing for Industry, Biomedicine and Art 3* (2020), 18:1–18:17. 1

[BKK19] Baek S., Kim K. I., Kim T.-K.: Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *IEEE/CVF CVPR* (2019), pp. 1067–1076. 2

[BKY97] Belhumeur P. N., Kriegman D. J., Yuille A. L.: The bas-relief ambiguity. In *IEEE CVPR* (1997), pp. 1060–1066. 5

[BTT*20a] Brahmbhatt S., Tang C., Twigg C. D., Kemp C. C., Hays J.: Contactpose: A dataset of grasps with object contact and hand pose. In *ECCV* (2020), vol. 12358, pp. 361–378. 2, 7, 8

[BTT*20b] Brahmbhatt S., Tang C., Twigg C. D., Kemp C. C., Hays J.: Contactpose: A dataset of grasps with object contact and hand pose. In *ECCV* (2020), vol. 12358, pp. 361–378. 2, 8

[CCY*21] Chen P., Chen Y., Yang D., Wu F., Li Q., Xia Q., Tan Y.: I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *IEEE/CVF ICCV* (2021), pp. 12909–12918. 2

[CG22] Chen T., Gu D.: Csa6d: Channel-spatial attention networks for 6d object pose estimation. *Cognitive Computation 14*, 2 (2022), 702–713. 2

[CKA*22] Christen S., Kocabas M., Aksan E., Hwangbo J., Song J., Hilliges O.: D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions. In *IEEE/CVF CVPR* (2022), pp. 20577–20586. 2

[CLM*21] Chen X., Liu Y., Ma C., Chang J., Wang H., Chen T., Guo X., Wan P., Zheng W.: Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *IEEE/CVF CVPR* (2021), pp. 13274–13283. 3

[CRKM21] Cao Z., Radosavovic I., Kanazawa A., Malik J.: Reconstructing hand-object interactions in the wild. In *IEEE/CVF ICCV* (2021), pp. 12397–12406. 1, 2

[CYX*21] Chao Y.-W., Yang W., Xiang Y., Molchanov P., Handa A., Tremblay J., Narang Y. S., Wyk K. V., Iqbal U., Birchfield S., Kautz J., Fox D.: Dexycb: A benchmark for capturing hand grasping of objects. In *IEEE/CVF CVPR* (2021), pp. 9044–9053. 2

[DHM*22] Du Z., Huang S.-S., Mu T.-J., Zhao Q., Martin R. R., Xu K.: Accurate dynamic SLAM using crf-based long-term consistency. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 28*, 4 (2022), 1745–1757. 1

[DNMC20] Doosti B., Naha S., Mirbagheri M., Crandall D. J.: Hope-net: A graph-based model for hand-object pose estimation. In *IEEE/CVF CVPR* (2020), pp. 6607–6616. 1, 2

[GHJK20] Garcia-Hernando G., Johns E., Kim T.-K.: Physics-based dexterous manipulations with estimated hand poses and residual reinforcement learning. In *IEEE/RSJ IROS* (2020), pp. 9561–9568. 1

[GRL*19] Ge L., Ren Z., Li Y., Xue Z., Wang Y., Cai J., Yuan J.: 3d hand shape and pose estimation from a single RGB image. In *IEEE/CVF CVPR* (2019), pp. 10833–10842. 2

[GTT*21] Grady P., Tang C., Twigg C. D., Vo M., Brahmbhatt S., Kemp C. C.: Contactopt: Optimizing contact to improve grasps. In *IEEE/CVF CVPR* (2021), pp. 1471–1481. 1, 2

[HHFS19] Hu Y., Hugonot J., Fua P., Salzmann M.: Segmentation-driven 6d object pose estimation. In *IEEE/CVF CVPR* (2019), pp. 3385–3394. 2, 3

[HLY*20] Hu S.-M., Liang D., Yang G.-Y., Yang G.-W., Zhou W.-Y.: Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences 63*, 12 (2020), 222103:1–222103:21. 8

[HOAL18] Höll M., Oberweger M., Arth C., Lepetit V.: Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *IEEE VR* (2018), pp. 175–182. 2

[HORL19] Hampali S., Oberweger M., Rad M., Lepetit V.: HO-3D: A multi-user, multi-object dataset for joint 3d hand-object pose estimation. *arXiv: 1907.01481* (2019). 2, 7

[HPSK21] Hwang J.-P., Park G., Suh I. H., Kwon T.: Primitive object grasping for finger motion synthesis. *Computer Graphics Forum 40*, 1 (2021), 266–278. 2

[HROL20] Hampali S., Rad M., Oberweger M., Lepetit V.: Honnotate: A method for 3d annotation of hand and object poses. In *IEEE/CVF CVPR* (2020), pp. 3193–3203. 2, 7

[HTB*20] Hasson Y., Tekin B., Bogo F., Laptev I., Pollefeys M., Schmid C.: Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *IEEE/CVF CVPR* (2020), pp. 568–577. 1, 2, 3, 4, 7, 8

[HVSL21] Hasson Y., Varol G., Schmid C., Laptev I.: Towards unconstrained joint hand-object reconstruction from RGB videos. In *International Conference on 3D Vision (3DV)* (2021), pp. 659–668. 2

[HVT*19] Hasson Y., Varol G., Tzionas D., Kalevatykh I., Black M. J., Laptev I., Schmid C.: Learning joint reconstruction of hands and manipulated objects. In *IEEE/CVF CVPR* (2019), pp. 11807–11816. 1, 2, 8, 11

[HYMH20] Huang J., Yang S., Mu T.-J., Hu S.-M.: Clustervo: Clustering moving instances and estimating visual odometry for self and surroundings. In *IEEE/CVF CVPR* (2020), pp. 2165–2174. 1

[HYZ*21] Huang J., Yang S., Zhao Z., Lai Y.-K., Hu S.-M.: Clusterslam: A SLAM backend for simultaneous rigid body clustering and motion estimation. *Computational Visual Media 7*, 1 (2021), 87–101. 1

[HZRS16] He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In *IEEE/CVF CVPR* (2016), pp. 770–778. 6

[IMB*18] Iqbal U., Molchanov P., Breuel T. M., Gall J., Kautz J.: Hand pose estimation via latent 2.5d heatmap regression. In *ECCV* (2018), vol. 11215, pp. 125–143. 2

[JLWW21] Jiang H., Liu S., Wang J., Wang X.: Hand-object contact consistency reasoning for human grasps generation. In *IEEE/CVF ICCV* (2021), pp. 11087–11096. 1

[KKB19] Kokic M., Kragic D., Bohg J.: Learning to estimate pose and shape of hand-held objects from RGB images. In *IEEE/RSJ IROS* (2019), pp. 3980–3987. 1, 2

[KKB20] KOKIC M., KRAGIC D., BOHG J.: Learning task-oriented grasping from human activity datasets. *IEEE Robotics and Automation Letters (RA-L) 5*, 2 (2020), 3352–3359. 1

[KMT*17] KEHL W., MANHARDT F., TOMBARI F., ILIC S., NAVAB N.: SSD-6D: making rgb-based 3d detection and 6d pose estimation great again. In *IEEE/CVF ICCV* (2017), pp. 1530–1538. 2

[KP06] KRY P. G., PAI D. K.: Interaction capture and synthesis. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 872–880. 2

[KP15] KIM J.-S., PARK J.-M.: Physics-based hand interaction with virtual objects. In *IEEE ICRA* (2015), pp. 3814–3819. 2

[KS12] KYOTA F., SAITO S.: Fast grasp synthesis for various shaped objects. *Computer Graphics Forum 31*, 2 (2012), 765–774. 2

[KUH18] KATO H., USHIKU Y., HARADA T.: Neural 3d mesh renderer. In *IEEE/CVF CVPR* (2018), pp. 3907–3916. 4

[KYZ*20] KARUNRATANAKUL K., YANG J., ZHANG Y., BLACK M. J., MUANDET K., TANG S.: Grasping field: Learning implicit representations for human grasps. In *International Conference on 3D Vision (3DV)* (2020), pp. 333–344. 1, 2

[LAZ*22] LI M., AN L., ZHANG H., WU L., CHEN F., YU T., LIU Y.: Interacting attention graph for single image two-hand reconstruction. In *IEEE/CVF CVPR* (2022), pp. 2761–2770. 1, 2, 5

[LF20] LI H., FAN L.: A flexible technique to select objects via convolutional neural network in vr space. *Science China Information Sciences 63*, 1 (2020), 1–20. 2

[LZXQ21] LI J.-C., ZHONG F., XU S., QIN X.: 3d object tracking with adaptively weighted local bundles. *Journal of Computer Science and Technology 36*, 3 (2021), 555–571. 2

[MDB*19] MUELLER F., DAVIS M., BERNARD F., SOTNYCHENKO O., VERSCHOOR M., OTADUY M. A., CASAS D., THEOBALT C.: Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 49:1–49:13. 3

[NNN20] NARASIMHASWAMY S., NGUYEN T., NGUYEN M. H.: Detecting hands and recognizing physical contact in the wild. In *NeurIPS* (2020), pp. 7841–7851. 2

[OKA12] OIKONOMIDIS I., KYRIAZIS N., ARGYROS A. A.: Tracking the articulated motion of two strongly interacting hands. In *IEEE CVPR* (2012), pp. 1862–1869. 3

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R. A., LOVEGROVE S.: Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE/CVF CVPR* (2019), pp. 165–174. 2

[PLH*19] PENG S., LIU Y., HUANG Q., ZHOU X., BAO H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In *IEEE/CVF CVPR* (2019), pp. 4561–4570. 2

[RIR15] ROGEZ G., III J. S. S., RAMANAN D.: Understanding everyday hands in action from RGB-D images. In *IEEE/CVF ICCV* (2015), pp. 3889–3897. 2

[RKI*14] ROGEZ G., KHADEMI M., III J. S. S., MONTIEL J. M. M., RAMANAN D.: 3d hand pose detection in egocentric RGB-D images. In *ECCV* (2014), vol. 8925, pp. 356–371. 2

[RKK09] ROMERO J., KJELLSTRÖM H., KRAGIC D.: Monocular real-time 3d articulated hand pose estimation. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (2009), pp. 87–92. 2

[RTB17] ROMERO J., TZIONAS D., BLACK M. J.: Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 245:1–245:17. 3

[SGSF20] SHAN D., GENG J., SHU M., FOUHEY D. F.: Understanding human hands in contact at internet scale. In *IEEE/CVF CVPR* (2020), pp. 9866–9875. 2

[SHCM21] SONG H.-X., HUANG J., CAO Y.-P., MU T.-J.: Hdr-net-fusion: Real-time 3d dynamic scene reconstruction with a hierarchical deep reinforcement network. *Computational Visual Media 7*, 4 (2021), 419–435. 2

[TBP19] TEKIN B., BOGO F., POLLEFEYS M.: H+O: unified egocentric recognition of 3d hand-object poses and interactions. In *IEEE/CVF CVPR* (2019), pp. 4511–4520. 1, 2

[TBS*16] TZIONAS D., BALLAN L., SRIKANTHA A., APONTE P., POLLEFEYS M., GALL J.: Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV) 118*, 2 (2016), 172–193. 2

[TG15] TZIONAS D., GALL J.: 3d object reconstruction from hand-object interactions. In *IEEE/CVF ICCV* (2015), pp. 729–737. 2

[TGBT20] TAHERI O., GHORBANI N., BLACK M. J., TZIONAS D.: GRAB: A dataset of whole-body human grasping of objects. In *ECCV* (2020), vol. 12349, pp. 581–600. 2

[WJW*20] WEI Z., JIANG R., WEI X., CHENG Y.-A., CHENG L., WANG C.: Novel indoor positioning system based on ultra-wide bandwidth. *Visual Computing for Industry, Biomedicine and Art 3* (2020), 1:1–1:6. 1

[WLLZ20] WANG M., LYU X.-Q., LI Y.-J., ZHANG F.-L.: VR content creation and exploration with deep learning: A survey. *Computational Visual Media 6*, 1 (2020), 3–28. 1

[WMB*20] WANG J., MUELLER F., BERNARD F., SORLI S., SOTNYCHENKO O., QIAN N., OTADUY M. A., CASAS D., THEOBALT C.: Rgb2hands: real-time tracking of 3d hand interactions from monocular RGB video. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 218:1–218:16. 3, 6

[WSH*19] WANG H., SRIDHAR S., HUANG J., VALENTIN J., SONG S., GUIBAS L. J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In *IEEE/CVF CVPR* (2019), pp. 2642–2651. 1, 3, 5

[XRW21] XU Z., RONG Z., WU Y.: A survey: which features are required for dynamic visual simultaneous localization and mapping? *Visual Computing for Industry, Biomedicine and Art 4* (2021), 20:1–20:16. 1

[YJLF22] YANG X., JIA X., LIANG Y., FAN L.: 6d object pose estimation in cluttered scenes from RGB images. *Journal of Computer Science and Technology 37*, 3 (2022), 719–730. 2

[YL12] YE Y., LIU C. K.: Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 41:1–41:10. 2

[YZL*21] YANG L., ZHAN X., LI K., XU W., LI J., LU C.: CPF: learning a contact potential field to model the hand-object interaction. In *IEEE/CVF ICCV* (2021), pp. 11077–11086. 1, 2, 3, 4, 6, 7, 8, 9

[ZBB21] ZAPPEL M., BULTMANN S., BEHNKE S.: 6d object pose estimation using keypoints and part affinity fields. In *Robot World Cup (RoboCup)* (2021), vol. 13132, pp. 78–90. 2

[ZHMW22] ZOU Z.-X., HUANG S.-S., MU T.-J., WANG Y.-P.: Objectfusion: Accurate object-level slam with neural object priors. *Graphical Models* (2022), 101165. 2

[ZLM*19] ZHANG X., LI Q., MO H., ZHANG W., ZHENG W.: End-to-end hand mesh recovery from a monocular RGB image. In *IEEE/CVF ICCV* (2019), pp. 2354–2364. 2

[ZSI19] ZAKHAROV S., SHUGUROV I., ILIC S.: DPOD: 6d pose object detector and refiner. In *IEEE/CVF ICCV* (2019), pp. 1941–1950. 2

[ZYSK21] ZHANG H., YE Y., SHIRATORI T., KOMURA T.: Manipnet: neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 121:1–121:14. 2

[ZZXW22] ZHAO Z., ZUO B., XIE W., WANG Y.: Stability-driven contact reconstruction from monocular color images. In *IEEE/CVF CVPR* (2022), pp. 1643–1653. 1, 2, 10, 11