

# CHORE：单张 RGB 图像的人-物交互重建

## 摘要

以往大多数关于从图像中感知 3D 人体的工作常在没有周围环境的情况下单独推理人类。然而，人类实际上是不断地与周围的物体相互作用的，所以重建模型不仅可以推理人类，还可以推理物体及其相互作用。但由于人与物体之间的严重遮挡、交互类型复杂和深度不确定性，这个推理问题实际上极具挑战性。在本文中，我们使用了 CHORE，一种学习从单个 RGB 图像中联合重建人和对象的新方法。CHORE 的灵感来自于隐式表面学习和经典基于模型的拟合的最新进展。CHORE 计算了两个无符号距离场，分别隐式表示的人和物体；以及一个对应参数化人体和物体姿态场。这些隐式场使我们能够稳健地拟合参数人体模型和 3D 对象模板，同时对他们的交互进行推理。此外，先前的像素对齐隐式学习方法在合成数据做出的假设在真实数据中并不适用。CHORE 提出了一种适用的深度感知缩放方法，允许对真实数据进行更有效的形状学习。CHORE 代码公布在：[github.com/xiexh20/CHORE](https://github.com/xiexh20/CHORE)。复现改进后的代码则在：[github.com/lnykyks/HOI\\_recon\\_ChorePaSta](https://github.com/lnykyks/HOI_recon_ChorePaSta)

**关键词：**人物交互；隐式学习

## 1 引言

为了在现实世界中部署机器人和智能系统，从视觉输入感知和理解人类与现实世界如何交互是必要的。虽然目前有大量关于从单个图像中重建 3D 人体的文献，但大多数作品孤立地处理人体。这些工作从大规模 3D 扫描中学习到了人体形状的丰富的先验信息，并使用运动学结构建模等方式来进行人体重建。目前关于基于图像进行人体与物体的联合重建的工作较少，仍处于起步阶段，这也是本文关注的重点。人与物体的联合重建极具挑战性：物体和人类相互遮挡，使得推理变得困难；深度的不确定性使得预测两者在 3D 世界中的相对大小和空间排列难以确定；且图像中关于交互的视觉信息有限。

在本项工作中，我们使用了 CHORE，一种学习从单个 RGB 图像中联合重建人和对象的新方法。CHORE 的灵感来自于隐式表面学习和经典基于模型的拟合的最新进展。CHORE 计算了两个无符号距离场，分别隐式表示的人和物体；以及一个对应参数化人体和物体姿态场。这些隐式场使我们能够稳健地拟合参数人体模型和 3D 对象模板，同时对他们的交互进行推理。此外，先前的像素对齐隐式学习方法在合成数据做出的假设在真实数据中并不适用。我们提出了一种适用的深度感知缩放方法，允许对真实数据进行更有效的形状学习。总的来说，CHORE 的主要贡献点在于：

- CHORE 是第一个端到端的从单张 RGB 图像重建人-物交互的方法。CHORE fields 的预测使得其能够将可控人体模型和物体模型拟合到图像中。
- 与之前使用弱透视相机并从合成数据中学习的工作不同, CHORE 使用全透视相机模型,这对于在真实数据上训练至关重要。因此 CHORE 也提出了一种新的训练策略, 其允许使用透视相机进行有效的像素对齐隐式学习。
- 经过有效的训练和联合重建, CHORE 的结果比现有技术有了 50% 的显著改进。

## 2 相关工作

### 2.1 单图像的人体重建

以往大多数重建的方法是基于统计 3D 身体模型, 这些模型从大规模 3D 扫描中学习到了人体形状的丰富的先验信息, 并使用运动学结构来对关节进行建模, 比如 SCAPE 模型 [3] 的相关工作。SMPL 模型使用姿态参数和形状参数描述人体, 前者描述人体的姿态, 后者描述人体各个 part 的形状。基于 2D 关节检测, SMPL 可以学习得到这两份参数, 从而得到经过姿态估计的 3D 人体 [26]。将 SMPL [26] 拟合到图像上的工作已经有了一些系列工作研究 [1, 2, 5, 10, 23, 29, 30]。但上述工作都孤立地看待人体, 我们的工作则增加了对人体、物体交互关系的考虑。

### 2.2 单图像的物体重建

给定一个图像, 3D 物体可以被重建为体素 [9, 16, 37]、点云 [13, 24] 或网格 [19, 31, 35]。与人类重建类似, 隐式函数在对象重建方面也取得了巨大的成功 [27, 28, 39]。这个方向的研究得到了大规模数据集 [33, 38] 的显着帮助。关于这些工作的回顾可以查阅 [15]。相比这些孤立重建物体的工作, 我们的方法可以联合重建出人体、物体以及他们的交互。

### 2.3 单图像的人物交互重建

早期方法主要通过观察人们随时间与场景的交互来推断 3D 几何, 比如 people watching [14] 的相关工作。近期方法采用了预先捕获的 3D 场景来推理三维人机交互, 比如人-物交互追踪的相关工作 [22]。近期方法中从 3D [7, 34]、2.5D [6, 8]、图像 [11, 12, 17, 20, 40] 等数据建模出手与对象交互, 性能令人印象深刻, 但局限于手, 没有涉及到整个人体, 比如从 grasp fields [21] 的相关工作。与我们相关的是早期方法, 因为用了对人体的感知。且我们预设的场景为野外, 不一定能够使用传感器、RGB-D 之类的方法构建场景。

与本项工作比较相似的是 [36, 41], 他们都孤立地从图像中重建出人体, 并利用场景信息或与物体的交互信息优化人体的姿势。PHOSA [41] 预定义了人和物体可能的交互方式, 进而根据图片推断实际的人-物交互。这种方法难以扩展且不一定准确。我们不依赖于这种启发式算法, 而是直接从数据中学习联合重建和交互先验。

### 3 本文方法

#### 3.1 本文方法概述

本文的任务是从单张 RGB 图像中重建出三维人体和物体网格，并考虑他们之间的接触。我们使用 SMPL 模型  $H(\theta, \beta)$  表示参数化人体，该模型将 3D 人体参数化为姿势  $\theta$  和形状  $\beta$  的函数，对于物体，则使用一个模板网格表示。

将三维人体和物体网格拟合到输入图片中是一个非确定解的问题，因此拟合的过程可能导致结果并不能很好地反映图像中人体和物体的姿态和关系。为此，我们首先从图像中重建出多个三维隐式场，作为网格拟合的目标。

图 1 展示了 CHORE 的整体流程，首先我们对图像进行实例分割，得到人体、物体的轮廓。之后我们将与轮廓堆叠后的 RGB 图像输入特征提取器  $f^{enc}$ ，预测出逐像素的特征，这些特征经过相机矩阵反投影到三维空间，然后与空间坐标拼接得到空间特征，再用于人/物体距离场、物体旋转及位移场的学习和预测，再分别用人体 SMPL 模板拟合人体距离场、用物体 mesh 模板经过旋转平移拟合物体距离场。

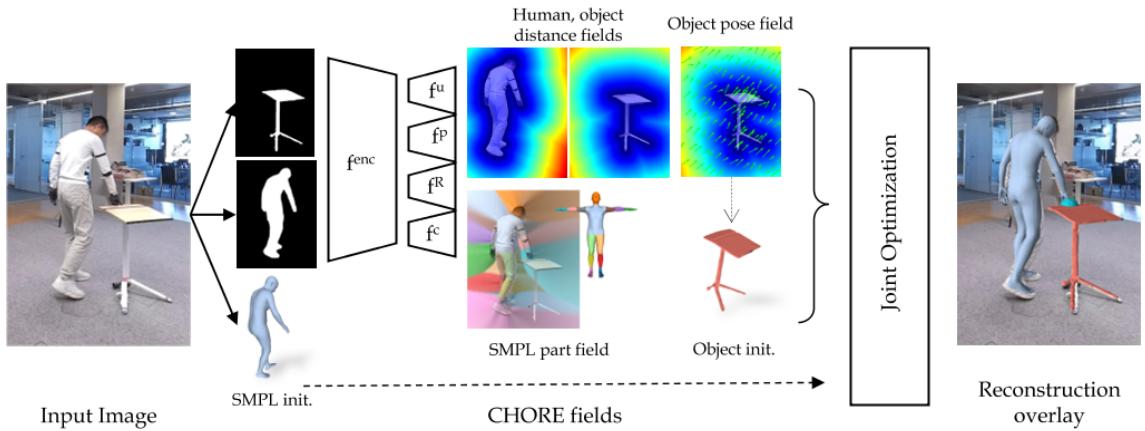


图 1. CHORE：联合重建过程

#### 3.2 特征提取模块

给定输入的 RGB 图像  $I$ ，我们首先进行 mask-rcnn [18] 实例分割，得到人体掩码  $p_{mask}$  和物体掩码  $o_{mask}$ ，然后我们将掩码  $p_{mask}, o_{mask}$  分别作为一个单独通道，堆叠到 RGB 图像  $I$  中。这里我们使用 Pifu [32] 作为图像的逐像素特征提取器  $f^{enc}$ ，输入即为堆叠了物体掩码的 RGB 图像。

#### 3.3 CHORE fields

**隐式表示.** 无符号距离场 (UDF) 是隐式表示的其中一种。首先简单介绍一下隐式三维表达。对于封闭的一个三维物体，我们需要获取其准确的形状时，可以通过一个函数获  $f$  得，我们输入一个三维坐标  $p$ ，函数输出一个点到物体表面的最近距离  $d$ ，那么所有满足  $f(p)=0$  的  $p$  就构成了物体的表面，那么这个函数  $f$  我们就可以称之为无符号距离场。

**UDF 解码器  $f^u$ .** 由于我们希望网络学习到一个空间无符号距离场，因此需要在空间中采样出点集  $P$ ，以人体无符号距离场为例，我们对点  $p \in P$  预测其到人体表面的无符号距离，

再与真实距离计算损失值。而仅仅使用  $p$  的坐标作为网络预测的信息来源是不充分的，本文的做法是将  $p$  根据相机参数投影到图像  $I$  所在的平面，从而可以得到  $p$  落在某个像素  $p_{pix}$ ，我们将该像素的特征拼接到  $p$  的三维坐标上，得到  $p$  的空间特征  $F_p$ ，再输入人体无符号距离场解码器  $f_h^u$ 。同理，通过这种方式构建的空间特征也用来预测其他隐式场。

**部件响应场解码器  $f_u^p$ .** 同理，我们也可以定义其他场，这是为了更好地重建出人体和物体而引入的其他约束。对于人体，我们最后是用 SMPL [26] 表示的参数化人体网格拟合到隐式场中，而仅仅是用隐式场可能导致拟合过程的歧义。因此，本文引入了一个部件响应场。具体来说，我们仍以人体为例，对空间中的点，我们通过  $f_u^p$  预测该点所属的人体部位。真实的人位是很容易得到的，SMPL 参数化人体本身就定义了人体的部位划分。

**旋转场解码器  $f^R$  和平移场解码器  $f^c$ .** 前文我们已经提到了，物体也构建了一个无符号距离场。然而，物体的隐式场构建则比人体的要复杂的多。我们通过输入图像和已有的人体姿态估计方法，可以很容易得到 SMPL 人体网格的初始姿态，所以从 SMPL 拟合到人体隐式场的过程几乎只需要简单位置移动的朝向调整即可。对于物体，选择与输入图像相似的物体网格本身就不简单，要得到其初始姿态则更加困难。因此，对于物体，我们除了要构建无符号距离场之外，还需要旋转场和平移场。简单来说，对于空间中的点，我们是用解码器  $f^R$  为其预测一个旋转矩阵，再将这些点的旋转矩阵求均值作为目标旋转矩阵，并与真实值计算损失值。平移场也是类似的思路，只不过把预测的目标从旋转矩阵调整为平移距离，对应的解码器则是  $f^c$ 。

### 3.4 损失函数定义

**隐式场训练损失项.** 对于人体和物体的无符号距离场解码器，我们在训练时先在真实数据表面附近采样得到点集  $P$ ，并计算出  $p \in P$  到表面的距离，作为预测的目标。这些采样点和  $f^{enc}$  输出的特征经过前文提到的拼接后，通过解码器得到预测的距离，并与真实值计算  $L_1$  距离，我们将人体和物体的损失项分别记为  $L_{u_h}$  和  $L_{u_o}$ 。对于部件响应场，我们用交叉熵作为损失项，记为  $L_p$ 。对于旋转场和平移场，我们用预测的旋转矩阵和平移距离与真实值计算均方误差，损失项分别记为  $L_R$  和  $L_c$ 。因此总的损失项可以记为：

$$L = \lambda_u(L_{u_h} + L_{u_o}) + \lambda_p L_p + \lambda_R L_R + \lambda_c L_c \quad (1)$$

**人体拟合损失项.** 对于人体拟合，我们用人体 UDF 场和部件响应场作为拟合的目标。从输入图像中我们先重建出 SMPL 人体网格的初始姿态，然后对网格顶点代入  $f^h$  得到预测的距离，我们优化人体使得该距离尽量小，即更加接近隐式表面。同时，我们也优化人体的姿态，使得部件响应场预测的部件与真实的部件尽量一致。我们将人体拟合损失项记为  $L_p$ 。从而可以得到人体拟合的损失项为：

$$E_{data}^h(\theta, \beta) = \sum_{p \in \Pi(\theta, \beta)} (\lambda_h \min(f_h^u(F_p), \sigma) + \lambda_{p'} L_p(l_p, f^p(F_p))), \quad (2)$$

**物体拟合损失项.** 我们使用了一个模板网格表示物体，将其记为  $O \in R^{3 \times N}$ ，旋转矩阵则记为  $R_o \in SO(3)$ ，平移记为  $t_o \in R^3$ ，大小则记为  $s_o \in R$ 。通过输入  $F_p$ ， $f^R$  预测出了旋转矩阵， $f^c$  则预测了平移与大小。那么变化后的物体记为  $O' = s_o(R_o O + t_o)$ ，然后我们将这样

一个调整姿态后的物体拟合到隐式场中，得到物体拟合的损失项为

$$E_{data}^o(R_o, t_o, s_o) = \sum_{p \in O'} (\lambda_o \min(f_o^u(F_p, \sigma)) + \lambda_{occ} L_{occ-sil}(O', M_o) + \lambda_{reg} L_{reg}(O')), \quad (3)$$

其中  $M_o$  表示物体的掩码,  $L_{occ-sil}$  表示物体投影轮廓与真实轮廓的遮挡损失。 $L_{reg}$  表示物体的正则化损失, 即  $O'$  中心与预测的中心的距离, 后者通过预测的平移值和 UDF 可以计算得到。

**联合优化损失项.** 除了人体拟合和物体拟合损失项外, 我们还需要考虑重建过程中人体和物体的交互信息, 这一损失项我们用  $E_{data}^c$  表示, 由于 UDF 等隐式场已经大致表示了人体和物体的相对位置, 那么交互可以简单定义为令人体网格和物体网格接触。为此, 我们使用倒角距离来调整物体的位置, 使得其能够与人体接触。

$$E_{data}^c(R_o, t_o, s_o) = \sum_{j=1}^K d(H_j^c(\theta, \beta), O_j^c), \quad (4)$$

在实际训练的过程中，上述三个能量项是一起训练的，即：

$$E_{data}(\theta, \beta, R) = E_{data}^h + E_{data}^o + E_{data}^c, \quad (5)$$

4 复现细节

图 2. CHORE 核心代码

图 2 展示本次复现代码的核心结构。代码量约 1000 行，包含了注释和笔者添加的功能代码 200+ 行。代码的 ReconFitterBase 类中大部分函数为基本的图形学算法函数，CHORE 的核心算法则集中在以下几部分。

损失函数.

compute-obj-loss 计算物体重建时的网格尺寸、旋转矩阵等损失。compute-prior-loss 则计算 SMPL 人体网格的先验损失，先验信息包含了人体的形状、骨骼结构等的约束。compute-df-h-loss 计算人体重建的 UDF 和真实值的 L1 距离。compute-contact-loss 先计算人体物体之间的部件距离是否小于阈值，是则用倒角距离优化这二者的位置。compute-collision-loss 计算人体和物体之间的碰撞损失。compute-kpts-loss 计算 3d 人体关节投影到 2d 后，与 2d 关节之间的距离损失。

#### 4.1 与已有开源代码对比

从 CHORE 的实现流程中可以看出，该项工作实现了人体和物体的拟合和重建，同时还实现了对人体和物体的交互的拟合和重建。但是该项工作在人体和物体的交互重建上并不关注和输入图像的匹配程度，仅仅使用倒角距离拉近物体和人体的距离可能导致物体和人体的交互并不合理。而本文的改进的重点在于引入了输入图像的信息，使得人体和物体的交互重建结果更加贴合输入图像。

#### 4.2 实验环境搭建

本次项目基于 Ubuntu 操作系统运行，使用 GPU:RTX3090 进行运算加速。

#### 4.3 图像交互信息获取

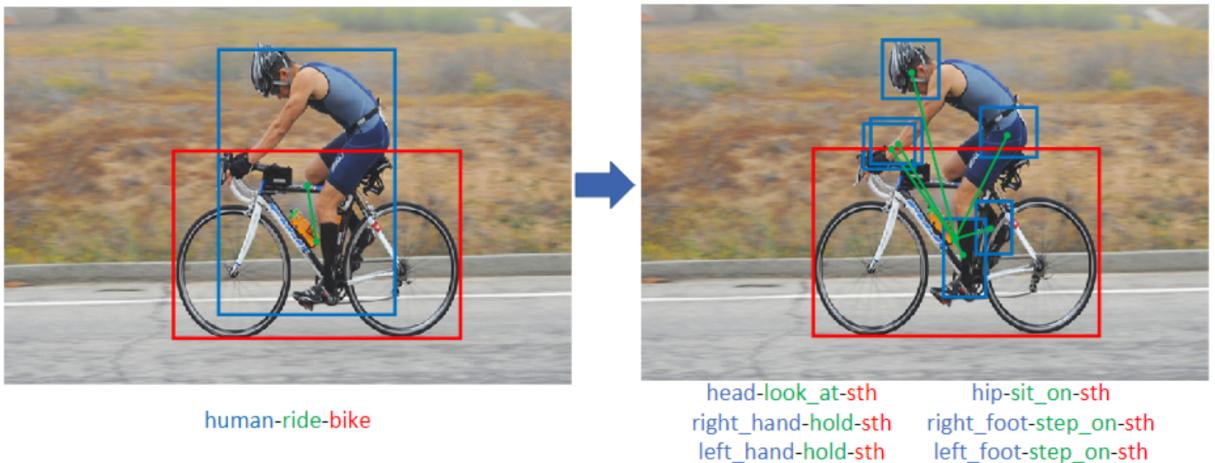


图 3. PaStaNet 预测结果

识别图像中的人-物交互信息有许多相关工作，比如 PaStaNet [25]，图 3其可以根据输入的图像提取出人体行为和每个部位的动作（即 part state）。使用 PaStaNet 的前提是，其预测的准确性和性能足够好。PaStaNet 训练数据包含 11.8 万张图片，其中有 28.5 万个人物、25 万个物体（与人物发生交互）、72.4 万个行为，这使得 PaStaNet 对人体部位动作的预测达到一个很高的精度。速度方面，PaStaNet 可以对 60FPS 的视频进行实时的人体部位动作输出，对图像的处理也就不在话下了，速度可以达到毫秒级。

由于我们使用 SMPL 参数化人体网格，而 SMPL 上各个顶点的顺序和所属的部位是固定的，这就为我们利用 PaStaNet 预测的人体部位动作提供了便利。根据 PaStaNet 的预测结果，我们可以得到哪些人体部位与物体发生交互，然后再优化物体姿态和位置，令物体与这些人体部位接触。这样优化之后，我们就可以得到和图片一致的人-物交互重建结果了。

#### 4.4 创新点

本次复现工作的主要代码实现创新点体现在两处，第一是将 PaStaNet 应用在 BEHAVE [4] 数据集上，得到每张中的人体部位动作，从而可以和同样在 BEHAVE 数据集上进行实验的

CHORE 进行最后的横向对比，第二是调整 CHORE 的损失函数，使得可以根据我们获得的人体部位动作，优化物体的位置和姿态。

图 4. 导出 PaStaNet 预测的人体动作

如图 5 所示，我们获取 PaStaNet 预测的初始结果，其中包含每个人体部位的动作和对应的置信度。但这些信息并不一定合理，也不一定能够完全和输入图像匹配。因此我们需要对 PaStaNet 的预测结果进行筛选。筛选的方法大致思路是：对于每个人体部位，我们保留置信度最高的动作，对于置信度低的动作，我们借助其他部位的动作以及先验信息修正该部位的动作，或将其置为“no interaction”。这样我们就得到了每张图像中与物体发生交互的人体部位和对应的动作。我们将结果存入到字典中并返回。

```

    def compute_collision_loss(self, smpl_verts, smpl_faces, obj_R, obj_t, 772
        obj_s, pasta=None): 773
        """ 774
        smpl_verts = torch.tensor(self.scan_v, dtype=torch.float32).repeat(obj_R, 775
        shape[0], 1, 1).to(self.device) 776
        verts = self.transform_obj_verts(smpl_verts, obj_R, obj_t, obj_s) 777
        ... 778
        person_mesh = pytorch3d.structures.Meshes(verts=smpl_verts, 779
        faces=smpl_faces.view(1, -1, 3)) 780
        obj_pc = pytorch3d.structures.Meshes(verts=obj_R, 781
        faces=obj_t, 782
        pen_loss = pytorch3d.loss.point_point_mse_distance( 783
        point_mesh, point_mesh.face_distance(person_mesh, obj_pc) 784
        ... 785
        # self.part_labels: (6809, ), value represent part index 786
        pasta_parts = { 787
            # hip: 788
            'hand': 2, # part index in self.part_labels 789
            'shoulder': 1, 790
            'elbow': 1, 791
            'rFoot': 6, 792
        } # part state to be checked 793
        if pasta is None: 794
            check_parts = list(pasta_parts.keys())
            for p in check_parts:
                if p not in pasta['o'].keys():
                    pasta['o'].update({p: 0})
            if not pasta['o'].containing:
                state = pasta['o'][p]
                if state not in self.states[p[1:]]:
                    pasta_parts.pop(p) # remove part that has no action 795
            # left pasta contains action 796
        if len(pasta_parts) > 0: # exists part(s) with action 797
            pen_loss = 0 798
            pen_losses = [] 799
            for part in pasta_parts: 800
                part_verts = smpl_verts[part] 801
                part_faces = smpl_faces[part] 802
                part_normals = smpl_normals[part] 803
                part_t = obj_t[part] 804
                part_R = obj_R[part] 805
                part_verts = self.transform_obj_verts( 806
                part_verts, part_R, part_t, None) 807
                object_mesh = trimesh.Trimesh( 808
                vertices=vert[0].detach().cpu().numpy(), 809
                faces=smpl_faces.detach().cpu().numpy(), 810
                process=False, 811
                validate=False, 812
                maintain_order=True, 813
                ) 814
                person_mesh = trimesh.Trimesh( 815
                vertices=vert[0].detach().cpu().numpy(), 816
                faces=smpl_faces.detach().cpu().numpy(), 817
                process=False, 818
                validate=False, 819
                maintain_order=True, 820
                ) 821
                cm = trimesh.collision.CollisionManager() 822
                cm.add_object('object', object_mesh) 823
                cm.add_object('person', person_mesh) 824
                result, contact_data = cm.in_collision_internal(return_data=True) 825
                for p in pasta_parts:
                    part_idx = pasta_parts[p]
                    part_mask = self.part_labels == part_idx
                    part_verts = smpl_verts[0][part_mask]
                    p_loss, _ = pytorch3d.loss.chamfer_distance(verts, part_verts, 826
                    part_normals, unqueeze(p), 827
                    pen_losses.append(p_loss) 828
                if len(pen_losses) > 1:
                    with torch.no_grad():
                        avg_loss = torch.mean(torch.stack(pen_losses))
                        remain = []
                        for idx in range(len(pen_losses)):
                            if pen_losses[idx] < avg_loss:
                                remain.append(idx)
                        for idx in remain:
                            pen_loss += pen_losses[idx]
            else:
                pen_loss += pen_losses[0]
            if result:
                return 0.02 / pen_loss
            else:
                return pen_loss
        else:
            # ! each time a pair of human and object
            object_mesh = trimesh.Trimesh( 829
            vertices=vert[0].detach().cpu().numpy(), 830
            faces=smpl_faces.detach().cpu().numpy(), 831
            process=False, 832
            validate=False, 833
            maintain_order=True, 834
            )
            person_mesh = trimesh.Trimesh( 835
            vertices=vert[0].detach().cpu().numpy(), 836
            faces=smpl_faces.detach().cpu().numpy(), 837
            process=False, 838
            validate=False, 839
            maintain_order=True, 840
            )
            cm = trimesh.collision.CollisionManager() 841
            cm.add_object('object', object_mesh) 842
            cm.add_object('person', person_mesh) 843
            result, contact_data = cm.in_collision_internal(return_data=True) 844
            pen_loss, _ = pytorch3d.loss.chamfer_distance(verts, smpl_verts)
            if result:
                if result:
                    return 0.02 / pen_loss
                else:
                    return pen_loss

```

图 5. 调整损失函数

如图 5 所示，我们将 CHORE 原本的损失函数 `compute_collision_loss` 更新。`pasta` 变量是一个字典，其中 `key` 是 SMPL 部位索引值，`value` 是对应部位的动作。考虑到 PaStaNet 预测的动作中有些并不是会发生交互的动作，比如手部的“point to”，“gesture to” 等动作，所以我们需要筛选。筛选之后我们得到 `pasta_parts` 这个字典，其包含了手部、脚部、臀部中符合目标动作的部位。若该字典为空，那么直接用整个人体网格和物体网格判断是否碰撞。若不空，则我们遍历每个字典中的人体部位，提取子网格，用来判断是否与物体发生交互，这里用到 `trimesh.Trimesh` 库的函数。在判断的过程中，我们也记录两者之间的倒角距离。由于 PaStaNet 预测的结果可能出现一定偏差，比如将无动作的手部预测成“carry”，所以对发生交互的人体部件，直接让物体去靠近这些人体部位可能导致不合理的结果。最常见的情况

是人的单手抓取一个物体，但 PaStaNet 预测左右手都是”carry”这一动作，直接让物体去靠近双手则出现物体在双手之间却不和其中任何一个接触的情况。因此对于我们记录的倒角距离，比较合适的做法是去除距离太远的那些部位，然后对剩下的部位进行倒角距离优化。

## 5 实验结果分析

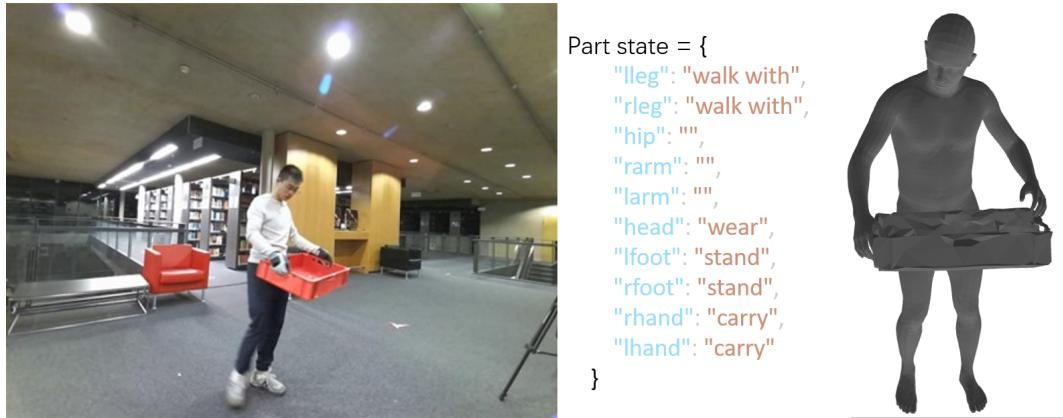


图 6. 上图左侧展示了输入图像，中间部分为 PaStaNet 预测的人体部位动作，右侧则为该图像中重建得到的 3D 人-物网格交互结果

我们最后使用的 CHORE 加入 PaStaNet 预测的信息后，在 BEHAVE 数据集上测试，测试类别共 8 个类别。

如图 6 所示，我们展示测试数据中的一个结果，从中我们可以看到 PaStaNet 基本上能够正确地预测出了人体部位动作。而以该动作表示的交互信息则很好地保证了重建结果中人体和物体交互的正确性：人的双手与物体都发生接触。

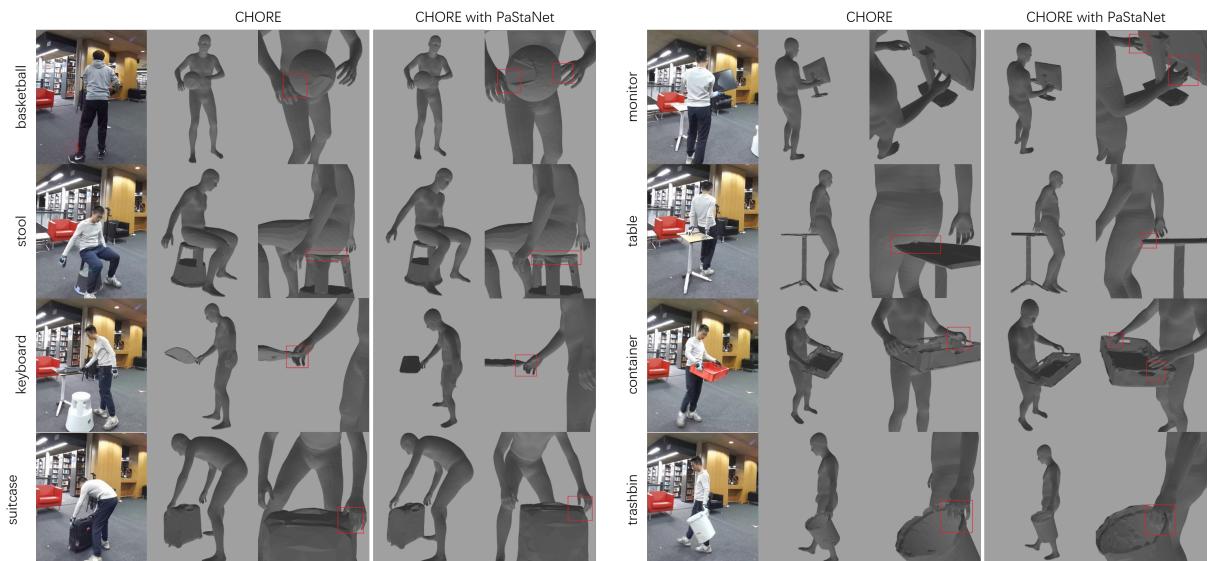


图 7. 实验结果对比。每行中图片的右侧两列表示 CHORE 的重建结果，第一列是人体和物体的重建结果，第二列则是交互细节的展示，有交互则用红框标出。第三第四列是加入 PaStaNet 后重建的结果，第三列展示整体，第四列展示交互细节。

图 7展示了我们的方法在 BEHAVE 的 8 个类别上的横向对比结果。可以看到，仅使用 CHORE 进行重建可能导致比较严重的穿模，而加入 PaStaNet 语义信息引导之后可以重建出穿模程度很小的结果。另外，未使用语义引导的情况下，物体如果隐式场预测不够准确，则会导致物体网格拟合之后姿态和输入图像偏差很大，这一点在图 7keyboard 类别的重建上有所体现。

## 6 总结与展望

总体上来看，加入了 PaStaNet 之后，CHORE 能够使人体和物体的交互重建更加贴合输入图像，在提升交互精度的同时，也能够尽量规避一些不合理的动作。但是 PaStaNet 预测的结果并不是完全准确的，这和模型训练的数据和预设动作类型有关。感知图像中交互信息的方式实际上还有很多其他工作，比如通过大语言模型，或者如今很成熟的以 Diffusion 为代表的二维图像处理的工作。

此外，使用可微 3D 表示和 Diffusion 优化的图像重建工作也出现了许多相关的工作，也是值得迁移到三维人-物交互重建的方法。

## 参考文献

- [1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018.
- [2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. *arXiv: Computer Vision and Pattern Recognition, arXiv: Computer Vision and Pattern Recognition*, Mar 2018.
- [3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005.
- [4] BharatLal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions.
- [5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. *Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image*, page 561–578. Jan 2016.
- [6] Samarth Brahmbhatt, Cusuh Ham, CharlesC. Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. *arXiv: Computer Vision and Pattern Recognition, arXiv: Computer Vision and Pattern Recognition*, Apr 2019.

- [7] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.
- [8] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. *ContactPose: A Dataset of Grasps with Object Contact and Hand Pose*, page 361–378. Jan 2020.
- [9] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. *3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction*, page 628–644. Jan 2016.
- [10] Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. Learned vertex descent: A new direction for 3d human model fitting.
- [11] Enric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Gregory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.
- [12] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. *arXiv: Computer Vision and Pattern Recognition, arXiv: Computer Vision and Pattern Recognition*, Mar 2020.
- [13] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.
- [14] David F Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A Efros, Ivan Laptev, and Josef Sivic. People watching: Human actions as a cue for single view geometry. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part V 12*, pages 732–745. Springer, 2012.
- [15] Kui Fu, Jiansheng Peng, Qiwen He, and Hanxiao Zhang. Single image 3d object reconstruction based on deep learning: A review. *Multimedia Tools and Applications*, page 463–498, Jan 2021.
- [16] Christian Hane, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, Oct 2017.
- [17] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

- [18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [19] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. *Cornell University - arXiv, Cornell University - arXiv*, Nov 2014.
- [20] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, Nov 2020.
- [21] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020.
- [22] Hedvig Kjellström, Danica Kragić, and Michael J Black. Tracking people interacting with objects. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 747–754. IEEE, 2010.
- [23] Nikos Kolotouros, Georgios Pavlakos, Michael Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.
- [24] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy J. Mitra, and Leonidas J. Guibas. Pix2surf: Learning parametric 3d surface models of objects from images. *Cornell University - arXiv, Cornell University - arXiv*, Aug 2020.
- [25] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Shiyi Wang, Hao-Shu Fang, Ze Ma, Mingyang Chen, and Cewu Lu. Pastanet: Toward human activity knowledge engine. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 382–391, 2020.
- [26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.
- [27] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.
- [28] Norman Muller, Yu-Shiang Wong, Niloy J. Mitra, Angela Dai, and Matthias Niesner. Seeing behind objects for 3d multi-object tracking in rgb-d sequences. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

- [29] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.
- [30] Gerard Pons-Moll and Bodo Rosenhahn. *Model-Based Pose Estimation*, page 139–170. Jan 2011.
- [31] Jhony K. Pontes, Chen Kong, Sridha Sridharan, Simon Lucey, Anders Eriksson, and Clinton Fookes. *Image2Mesh: A Learning Framework for Single Image 3D Reconstruction*, page 365–381. Jan 2019.
- [32] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019.
- [33] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.
- [34] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. *GRAB: A Dataset of Whole-Body Human Grasping of Objects*, page 581–600. Jan 2020.
- [35] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. *Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images*, page 55–71. Jan 2018.
- [36] Zhenzhen Weng and Serena Yeung. Holistic 3d human and scene mesh estimation from single view images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.
- [37] Jiajun Wu, Yifan Wang, Tao Xue, Xiaobin Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5d sketches. *Neural Information Processing Systems, Neural Information Processing Systems*, Jan 2017.
- [38] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. *ObjectNet3D: A Large Scale Database for 3D Object Recognition*, page 160–176. Jan 2016.
- [39] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Cornell University - arXiv, Cornell University - arXiv*, May 2019.

- [40] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021.
- [41] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. *Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild*, page 34–51. Jan 2020.