Durham University

Durham E-Theses

3D Representation Learning for Shape Reconstruction and Understanding

YU, ZHENGDI

How to cite:

YU, ZHENGDI (2023) *3D Representation Learning for Shape Reconstruction and Understanding*, Durham theses, Durham University. Available at Durham E-Theses Online: http://etheses.dur.ac.uk/15105/

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the full Durham E-Theses policy for further details.

Academic Support Office, The Palatine Centre, Durham University, Stockton Road, Durham, DH1 3LE
e-mail: e-theses.admin@durham.ac.uk Tel: +44 0191 334 6107
http://etheses.dur.ac.uk

3D Representation Learning for Shape Reconstruction and Understanding

Zhengdi Yu

A thesis presented for the degree of
Master by Research

Department of Computer Science
Durham University
United Kingdom
June 2022

Abstract

The real world we are living in is inherently composed of multiple 3D objects. However, most of the existing works in computer vision traditionally either focus on images or videos where the 3D information inevitably gets lost due to the camera projection. Traditional methods typically rely on hand-crafted algorithms and features with many constraints and geometric priors to understand the real world. However, following the trend of deep learning, there has been an exponential growth in the number of research works based on deep neural networks to learn 3D representations for complex shapes and scenes, which lead to many cutting-edged applications in augmented reality (AR), virtual reality (VR) and robotics as one of the most important directions for computer vision and computer graphics.

This thesis aims to build an intelligent system with dynamic 3D representations that can change over time to understand and recover the real world with semantic, instance and geometric information and eventually bridge the gap between the real world and the digital world. As the first step towards the challenges, this thesis explores both explicit representations and implicit representations by explicitly addressing the existing open problems in these areas. This thesis starts from neural implicit representation learning on 3D scene representation learning and understanding and moves to a parametric model based explicit 3D reconstruction method. Extensive experimentation over various benchmarks on various domains demonstrates the superiority of our method against previous state-of-the-art approaches, enabling many applications in the real world. Based on the proposed methods and current observations of open problems, this thesis finally presents a comprehensive conclusion with potential future research directions.

Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and is all the original work of the candidate with the exception of where collaboration is explicitly stated within the thesis text.

Copyright © 2022 by Zhengdi Yu.

"The copyright of this thesis rests with the author. No quotations from it should be published without the author's prior written consent and information derived from it should be acknowledged".

Acknowledgements

First of all, I would like to thank my supervisor, Professor Toby Breckon. for his appreciation, kindness and remarkable supervision. His great expertise and extreme support are the best boosts for this research journey. He is absolutely the greatest supporter and best supervisor for me. Having him as my Master by Research supervisor is the greatest choice I have made in my academic life.

Second, I would like to thank all the labmates, colleagues and co-authors I met within the great team, for their research insights, support and companionship. As maybe the youngest person in the team and the only master's student, I have always been learning from them.

Most importantly, being Mrs. Yuqiong Tian's son is the greatest honour of my life. A "Thank you" or any other cliche will never be able to perfectly express the feelings. I believe there would always be another universe where she can still support me and love me the best in the world, where she could witness my graduation.

Contents

List of Figures

List of Tables

Dedication

To Mrs. Yuqiong Tian, the best mom in the world.

CHAPTER 1

Introduction

1.1 Motivation

Building up dynamic 3D representation is essentially building up a digital copy of the real world - characters, settings, plots are the three main factors of the event (Figure 1.1). "Translated" into a computer science style, they are **creatures (humans, animals), scenes (objects, time and locations) and motions/interactions**, where the first two factors are the most important. However, it poses various challenges for researchers in many research realms of computer vision and computer graphics such as: 3D shape reconstruction and generation, neural rendering, scene understanding, human pose estimation and shape recovery, semantic and instance segmentation, and intersections such as HCI and human-scene interaction. These works has enabled various applications for virtual reality (VR), augmented reality (AR) and robotics.

In terms of representing objects and scenes, most classical early approaches rely on hand-crafted features or strong geometric priors. Classical approaches for scene reconstruction mainly include SfM (structure from motion) [16, 17] and SLAM (simultaneous localization and mapping) [18, 19]. However, they are agnostic to the

Figure 1.1: **Motivating example:** The key factors of building a virtual world are: humans, scenes and interactions. The first two factors are the most important.

objects and composition of the scene. Learning-based methods for 3D shape representation has been extensively studied in recent years, and they can be categorized into explicit representations and implicit representations by the format of their output. Moreover, there are many richly annotated 3D datasets such as: ScanNet [5], S3DIS (2D-3D-S) [12], SceneNN [11], ShapeNet [20], and popular outdoor benchmarks such as SemanticKITTI [21], and famous parametric human and hand benchmarks InterHand2.6M [13] and 3DPW [22]. Benefiting from them, there is an exponential growth in the number of research works pushing forward the development of 3D representation learning.

Explicit methods mainly include voxel representations, mesh representations and point cloud representations. They are commonly used for representing shapes with pre-defined topologies, especially, the **mesh representations** are natural for representing **humans**, but there is not yet a powerful representation for the most difficult and flexible part of human - hands, especially interacting hands. Despite the great and promising performance, it is well-known that explicit methods are typically limited to the topologies of their representations when extended to **scene-level** rep-

resentations, such as 1) memory usage, or resolution in voxel-based representations, 2) fixed and limited number of points in point representation, and 3) limited template and single topology in mesh representation. As a result, it is difficult for them to be efficiently adapted to large-scale applications such as scene representation. To address the discretization issue of explicit representations, implicit function learning (IFL) methods leverage multi-layer perceptrons (MLPs) to learn an implicit function to represent the 3D continuous surface. However, all the existing methods still suffer from open problems and unnecessary constraints of input topology. Signed distance filed (SDF) and occupancy field (OF) are commonly used for implicit function encoding from single-view images or point clouds or occupancy grids. After that, unsigned distance field (UDF) [6] and neural radiance field (NeRF) [23] present amazing performance, However, the voxelization process and the ambiguous interpolation problem cannot be well addressed so far, there are also plenty of open problems for NeRF pipelines, such as the lack of geometric constraints on the underlying volume rendering.

Despite the amazing and promising performance and experimentation of the existing 3D representation learning, there is not yet a perfect 3D representation for efficient real-world application and considerable practicality. As the main motivation of this thesis, **we aim to explore and build up both powerful explicit 3D representations and implicit 3D representations from the real-world data, such as point clouds and images, for the two factors - scenes and humans separately**, and build the intelligent system with powerful 3D representations for shape representation and scene understanding, serving for future application and research.

1.2 Publications and Contributions

In this section, I will summarize the contributions to each of the works contributed to the thesis. During the period of MRes, I was the main contributor in the following research publications:

- **P2-Net: Joint Description and Detection of Local Features for Pixel**

and Point Matching, B. Wang, C. Chen, Z. Cui, J. Qin, C. X. Lu, Z. Yu, P. Zhao, Z. Dong, F. Zhu, N. Trigoni, et al, In Proceedings of the IEEE/CVF International Conference on Computer Vision (**ICCV**), **2021**. [24]

- **RangeUDF: Semantic Surface Reconstruction from 3D Point Clouds**, B. Wang, Z. Yu, B. Yang, J. Qin, T.P. Breckon, L. Shao, N. Trigoni, and A. Markham., arXiv preprint arXiv:2204.09138, **2022**. [25] (Contributing to Chapter 3)

- **ACR: Attention Collaboration-based Regressor for Arbitrary Two-hand Reconstruction**, Z. Yu, S. Huang, C. Fang, T.P. Breckon, and J. Wang, In Conference on Computer Vision and Pattern Recognition (**CVPR**), **2023** [26] (Contributing to Chapter 4)

For clarity, regarding my independent contribution: In each of the above works, I did coding, experiments and visualization as a **main** contributor for joint works, and I also led the first-authored papers, which includes coding, experiments, visualization and writing. However, I only include the most relevant and representative works in this thesis for representation learning. In addition to the research outputs, I have also put my research into practice and finished some applications such as [27] demonstrated in Figure A.2 of **Appendix**, which is a motion capture system developed based on UE5 [28] and Blender [29].

1.3 Thesis Structure

This thesis is organised as below:

- Chapter 2 presents a comprehensive literature review of representation learning for shape reconstruction and understanding. In this chapter, learning-based representation learning methods are categorized by the output format of their methods. Both explicit representation and implicit representations are studied and included in this chapter and the thesis.

- Chapter 3 explores implicit 3D representation learning and introduces our RangeUDF, a new implicit representation method to recover the geometry

and semantics of continuous 3D scene surfaces from sparse raw 3D point clouds. Compared with existing approaches, our method is the first one to directly reconstruct 3D semantic surfaces from sparse point clouds. Thanks to the proposed range-aware neural unsigned distance fields and surface-oriented segmentation module, we explicitly address the open problem of surface ambiguity and can infer high-quality semantic classifications for the implicit surface. Extensive experiments show that we outperform the previous state-of-the-art methods [6, 30] by a large margin with much less memory and time consumption. Moreover, our method shows superiority in bridging the gap between real-world data and synthetic data to generalize even cross-domain and across unseen datasets.

- Chapter 4 explores explicit 3D representations learning and presents our ACR, which makes the first attempt to reconstruct hands in arbitrary scenarios with superior performance and least constraints with a 3D parametric model MANO [3], taking only the raw monocular RGB image as input without any external detector in a one stage manner. ACR achieves promising results by representation disentanglement and attention aggregation. We extensively evaluate our method on various types of hand reconstruction datasets and conduct experiments on in-the-wild videos and images. We demonstrate that our method significantly outperforms the best interacting-hand approaches on quantitative results, qualitative results, and practicality for applications in AR and VR.

- Chapter 5, the final chapter, summarizes the key contributions of the thesis and provides a discussion on the limitations and future works of 3D representation learning.

CHAPTER 2

Literature Review

In this section, we discuss the prior work in the literature related to 3D representation learning and shape reconstruction and understanding. Reconstruction of geometric and semantic information are two fundamental tasks for real-world understanding. 3D shape reconstruction has been studied for decades. Classical methods mainly include Structure-from-Motion (SfM) [16, 17] and Simultaneous Localization and Mapping (SLAM) [18, 19] but they are often agnostic to objects and instance information as introduced in the comprehensive survey [31]. Moreover, due to the reliance on 2D information, we are inherently unable to recover occluded regions without geometric priors and the reconstructions are often sparse point clouds.

In this thesis, we focus on learning-based methods, which can be broadly categorized into explicit representations and implicit representations methods by their output format. It has been shown that implicit representations have great potential for representing complex shapes whilst explicit representations are natural to represent topologies with pre-defined templates such as humans and animals. We explore both explicit representations of humans and implicit representations of scenes in the thesis. This section summarizes the related works of these two representations.

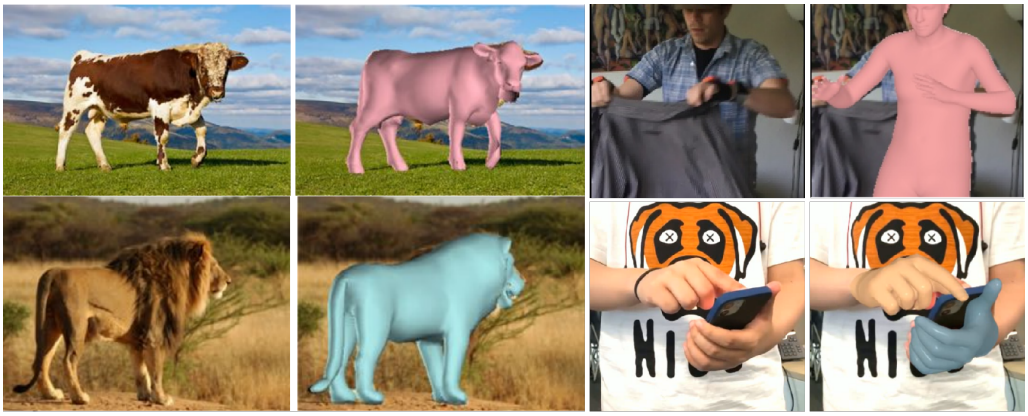


Figure 2.1: **Explicit representations** are natural for representing 3D shapes with a single topology or template such as human, animal, corresponding to SMPL/SMPL+X/MANO model [1–3] and examples from SMAL model [4] (left).

2.1 Explicit 3D Representation

To explicitly model 3D geometry of scenes and objects from both images or point clouds, voxel grids [32], octree [33], point clouds [34], triangle meshes [35, 36], and shape primitives [37] are widely used in many impressive prior arts. In spite of the great performance they have achieved in scene understanding [38, 39], shape reconstruction [40, 41], completion [42], and shape generation [43], such discrete 3D shape representations are inherently limited by the resolution and memory cost. Nevertheless, among them, mesh representation with the pre-defined template is the most natural pipeline for representing objects with certain shapes and regular variances. In the following paragraphs, we further discuss the **mesh representation** with parametric models to represent humans.

(1) Parametric 3D Human Reconstructions: is a branch of explicit representations for 3D shape reconstruction. Parametric human body models such as SMPL [1] have been widely adopted to encode the complex 3D human mesh into a low-dimensional parameter vector. Some existing methods [36, 44–47] have achieved impressive performance using various weak supervision signals, such as 2D pose annotations [48–50], semantic segmentation [51], temporal information [52, 53] and motion dynamics [54], optimization [55] in the loop [56] and geometric priors[49], etc. Along with the development of RGB-based hand reconstruction works [8, 26, 57, 58], the body model is finally integrated with a statistical hand model MANO [59] and a

face model to form SMPL-X [2], which can tackle holistic reconstruction problems. Another pipeline of optimization-based methods such as SMPLify-X [2] fits SMPL-X [2] to 2D body, hand and face keypoints [60] estimated in an image. However, there are still many problems for parametric human reconstruction, such as instability under occlusion, agnostic to camera trajectory and pose, and reliance on hand detector and bounding-box-level feature to recover two hands reconstruction.

(2) Two-Hand Reconstruction: A straightforward way to deal with two-hand reconstruction is to locate each hand separately and then transform the task into single-hand reconstruction. This strategy is commonly adopted in full-body reconstruction frameworks [61–66]. However, independently reconstructing two hands remains a failure in addressing interacting cases, as the close interacting hands are usually inter-occluded and could easily confuse the prediction. Earlier works mainly dealt with hand interaction relying on model fitting, multi-view or depth camera setup. For instance, Taylor et al. [67] introduced a two-view RGBD capture system and presented an implicit model of hand geometry to facilitate model optimization. Mueller et al. [68] simplified the system by using only a single depth camera. They further proposed a regression network to predict segmentation masks and vertex-to-pixel correspondences for pose and shape fitting. Smith et al. [69] adopted a multi-view RGB camera system to compute keypoints and 3D scans for mesh fitting. To handle self-interaction and occlusions, they introduced a physically-based deformable model that improved the robustness of vision-based tracking algorithms.

Recent interest has shifted to two-hand reconstruction based on a single RGB camera. Wang et al. [9] proposed a multi-task CNN that predicts multi-source complementary information from RGB images to reconstruct two interacting hands. Rong et al. [70] introduced a two-stage framework that first obtained initial prediction and then performed factorized refinement to prevent producing colliding hands. Similarly, Zhang et al. [14] predicted the initial pose and shape from deeper features and gradually refined the regression with lower-level features. The latest work [8] introduced a GCN-based mesh regression network that leverages pyramid features and learned implicit attention to address occlusion and interaction issues. However, these methods primarily treat two hands as an integral and implicitly learn an en-

tangled representation to encode two hands with bounding-box-level features and are only applicable for closely interacting scenarios.

2.2 Implicit Representation and Reconstruction

To address the discretization and resolution issues of explicit representations, implicit methods leverage multi-layer perceptrons (MLP) to learn an implicit function to represent the 3D continuous surface, and directly infer outputs from the continuous input space with more memory-efficient shape representations.

(1) Implicit Function Learning (IFL): Specifically, implicit representations can be generally categorized as: 1) Occupancy field (OF) [71, 72], 2) signed distance field (SDF) [73], 3) unsigned distance field (UDF) [6, 74], 4) neural radiance fields (NeRF) [23], and 5) hybrid fields [75]. Among them, occupancies and signed distance fields have achieved promising results in representing simple 3D shapes such as chairs, cars or sofas. However, they can only represent closed 3D shapes. Instead, approximate SDF can have sign error and non-constant derivatives yet still be marchable by reducing the step size proportional to the error in the SDF derivative. Applying deformation such as undergoing twisting or stretching to an exact SDF results in non-exact SDF. The non-watertight mesh can be extracted by computing the distance to the nearest triangle and using the normal to determine inside/outside. As the second pipeline of implicit function learning methods, NeRF series [23] focus on neural rendering and usually can not produce high-quality reconstructions due to the lack of geometric constraints. In the past two years, some implicit function learning methods also achieved impressive results in the areas of 1) 3D shape reconstruction [76, 77], 2) 3D shape generation [78], 3) novel view synthesis [79] and 4) scene understanding [80], but most of these approaches are image-based, and almost all of them only focus on objects or small-scale scenes or rely on the inefficient sliding window approach to recover the entire scene.

(2) Neural Implicit Surface Reconstruction: Early approaches use an encoder [71, 72] or an optimization-based process to vectorize the 3D shape into a latent code and decode the shape into surface reconstructions with a decoder. With

Method Type	(a) Closed surfaces	(b) Open surfaces	(c) Functions/Manifolds	(d) Complex scenes
SDF/OF	✔	✗	✗	✗
Approximate SDF	✔	✔	✗	✗
UDF	✔	✔	✔	✔

Table 2.1: **Implicit representations** are more flexible with more properties and practical for real-world applications with complex geometry and topologies such as 3D scenes.

the development of implicit function learning (IFL), there have been more advances in the 3D surface reconstruction area with neural implicit functions such as the occupancy field (OF), signed distance field (SDF), and unsigned distance field (UDF). Most existing approaches such as [30, 81–84] rely on either binary occupancies or signed distance fields to represent the implicit 3D surfaces and use Marching Cubes algorithm [85] to recover the implicit surface into full mesh geometry. However, SDF and occupancies can only represent closed surfaces with limited topologies. Among them, NDF [6] learns an implicit function of unsigned distance fields for continuous surface reconstruction with more topologies. However, it relies on a time-consuming sliding window strategy and memory-consuming voxelization. All of these methods can only deal with small-scale scenes or object-level reconstructions and they all suffer from the surface ambiguity problem due to trilinear interpolation.

(3) 3D Semantic Segmentation and Panoptic Segmentation: To learn semantic information for point clouds, existing approaches mainly include 1) projection and voxel-based methods [86] and 2) point-based methods [7, 87]. In addition, Kirillov et al. [88] proposed to unify semantic and instance segmentation, which they termed **panoptic segmentation**, together with an evaluation metric, the panoptic quality (PQ). Although they have achieved excellent semantic segmentation accuracy, these methods are essentially designed for the discrete 3D points from point clouds instead of dense scene surfaces. Despite the fast development of implicit 3D scene representations, no prior art can infer semantic information for the implicit scenes from sparse real-world point clouds.

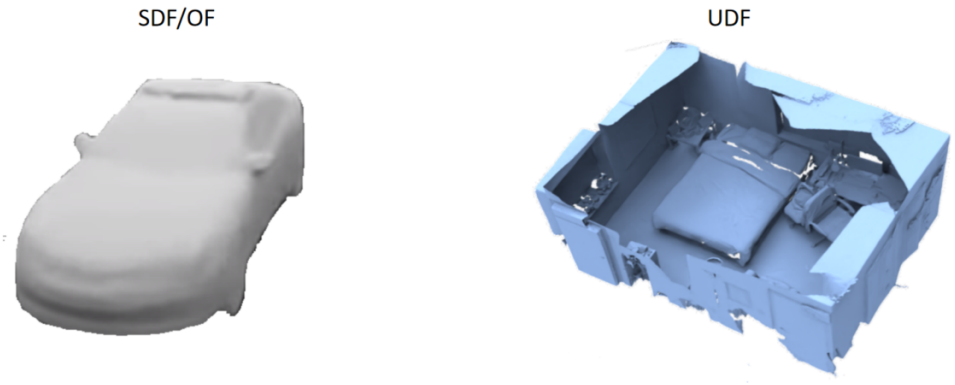


Figure 2.2: Comparison between different implicit representation methods. UDF can present open surfaces with complex topologies (right), while SDF or OF can only represent closed surfaces and object-level meshes such as car without inner structures (left).

2.3 Summary

First of all, for the scene representations with implicit function learning methods from point clouds, the previous state-of-the-art methods are [6, 30, 89]. They have achieved considerable results on simple and synthetic data. However, due to the use of simple trilinear interpolation, they all suffer from the surface ambiguity problem, resulting in over-smooth surface reconstruction and loss of details. Furthermore, it is difficult to extend their works to real-world applications with semantic information since they rely on slow voxelization and a sliding window strategy. In contrast, we explore a novel strategy with unsigned distance fields to efficiently recover the precise surface geometry without suffering from surface ambiguity and we are no longer limited to any type of topologies. Moreover, we design a surface-oriented semantic segmentation module to propagate the semantic classifications to the implicit surface for further applications.

Secondly, for the interacting two hands reconstruction with explicit representation learning, the previous state-of-the-art methods are IntagHand [8] and InterShape [14]. They typically adopt bounding-box-level features with an external detector to encode the two hands as an integral to learn an entangled representation. As a result, their methods fail to handle the occlusion, and truncation and

they are essentially only tailored for one scenario of closely interacting hands. These drawbacks highly limit their practicality in applications. Observing these open problems, we design the first one-stage method to handle two-hand reconstruction under arbitrary scenarios by representation disentanglement and attention collaboration.

CHAPTER 3

Neural Implicit Representation Learning for Scene Understanding

Reconstructing continuous surfaces from sparse or incomplete 3D point clouds is a fundamental problem in computer vision, computer graphics, and robotics vision. However, explicit shape representation has been studied for decades, it is known that these methods are often limited to single topologies such as voxel, point cloud or mesh and there are many drawbacks in computation efficiency and memory costs. In that, a growing number of implicit function learning (IFL) methods had been developed for 3D reconstruction and shape representation. Compared to explicit representations, implicit functions are more natural to represent more topologies and can output continuous and complex surfaces, manifolds, or functions.

In this chapter, we introduce RangeUDF [25], a new implicit representation based method to recover the geometry and semantics of continuous 3D scene surfaces from sparse raw 3D point clouds produced by sensors like LIDAR or RADAR. Unlike occupancy field (OF) or signed distance field (SDF) which can only represent closed surfaces, our approach is not limited to any type of topologies. As one of our main contributions, being different from the existing unsigned distance fields based methods, our framework does not suffer from any surface ambiguity. In addition,

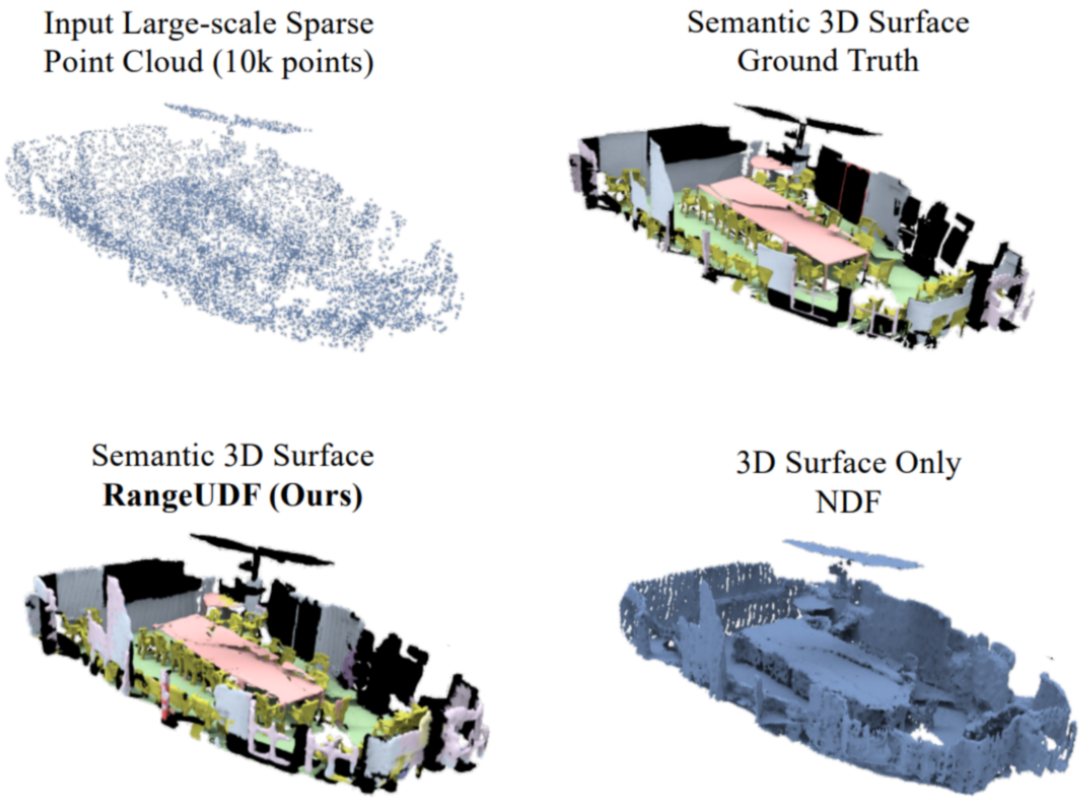


Figure 3.1: **Motivating example**: Given a sparse input point cloud with complex structures from ScanNet [5], RangeUDF simultaneously recovers precise geometry and semantic information of continuous 3D surfaces, while existing methods such as NDF [6] cannot.

our RangeUDF can jointly predict accurate semantic labels for the implicit continuous surfaces in a surface-oriented manner. The key idea to our approach is 1) a range-aware unsigned distance fields function together with a 2) surface-oriented semantic segmentation module, and 3) an efficient point-wise encoder. Extensive experiments show that our RangeUDF clearly outperforms previous state-of-the-art approaches in surface reconstruction on four different point cloud datasets. Moreover, RangeUDF is the first one that can bridge the gap between synthetic data and real-world data, which presents superior generalization ability on unseen data across multiple datasets.

3.1 Introduction

Recovering fine-grained geometry and precise semantic information of a 3D scene is a fundamental and important research problem for many cutting-edge applications such as augmented reality (AR), virtual reality (VR), and home robotics. Classical approaches such as Poisson surface reconstruction [90] often rely on strong geometric priors such as local linearity, which will cause the reconstruction to be over-smooth and lose fine-grained details.

In terms of integrating 3D semantics, current 3D semantic mapping is typically achieved by associating semantic labels with geometric representations generated from various 3D reconstruction approaches. Among them, learning-based methods can be broadly categorized by their output representation as explicit or implicit. In this chapter, we focus on implicit 3D representations. Typical implicit representation methods encode geometries into multilayer perceptions (MLP). Recent research in implicit methods has shown their potential in representing complex 3D shapes from either images or point clouds and manifolds. Implicit representations can also be categorized as 1) signed distance field (SDF) [81], 2) occupancy field (OF) [30, 72], 3) neural radiance field (NeRF) [23], and 4) hybrid representation [91]. These works had achieved promising results in the areas of 1) 3D shape reconstruction [76, 77], 2) 3D shape generation [92], 3) novel view synthesis [93] and 4) scene understanding [80]. However, most of these works are image-based and few works can reconstruct complex 3D scenes with semantic information from raw sparse point cloud data from sensors. Essentially, this is due to the drawback of their representations. They simply cannot be adapted to represent such kind of open surface formed by point clouds with arbitrary topologies.

As a result, the main issue of these works is the lack of ability to model arbitrary types of topologies such as open surfaces, which severely constrains their practicality of them. Specifically, SDF and occupancies can only represent open surfaces. NeRF can also take point clouds as input, however, due to the lack of geometric constraints caused by the underlying volume rendering, they can only produce rough surfaces. To address these issues, there are several concurrent works [6, 74]. Among them, SAL [74] does not require closed data for training, however, the output represen-

tation is again SDF, therefore still can only represent closed surface with no inner structure as well. NDF [6] takes occupancy grid as input for shape encoding, and subsequently predicts unsigned distance fields with point feature queried with trilinear interpolation and it presents promising small-scale object reconstruction results on open surfaces with inner structures such as cars.

Despite the promising small-scale object-level reconstruction results of NDF, it cannot be efficiently extended to scene-level reconstruction inherently due to the reasons below: 1) Limitations of voxel-based representations. It relies on time-consuming voxelization and sliding window strategy to process the data and query local features for discrete query points with limited voxel resolution. Moreover, it requires a slow sliding window strategy with high computational cost to cut the individual scene into small cubes and apply NDF on each of them. This will inevitably cause the loss of fine-grained details, and 2) losing the awareness of the integral and correct 3D geometry (e.g., a chair or a bed is cut into two different cubes.), which also makes it difficult to integrate instance or semantic information for further applications. 3) Finally, NDF and other prior arts typically adopt naive trilinear interpolation for feature querying from the nearest neighbours that lie in the vicinity of the voxel. However, this strategy will lead to surface ambiguity and over-smooth surface reconstruction as shown in the following method and experiments sections.

In this chapter, we present **RangeUDF**, a **range**-aware neural implicit representation with **u**nsigned **d**istance **f**ields to recover the precise reconstruction of the continuous implicit surface geometries and semantics from large-scale, raw and sparse point clouds without suffering from the limitations mentioned above. Specifically, our framework consists of four modules: 1) a per-point feature extractor to efficiently process large-scale point cloud, 2) a neighbourhood searching module for fast K nearest neighbours searching of the query point, 3) a range-aware unsigned distance fields implicit function with neural interpolation module, and 4) an implicit surface-oriented semantic segmentation module to infer semantic information for the underlying implicit surface.

As shown in Figure 3.1, built on these components, we are the first one to recon-

struct 3D semantic surfaces directly from sparse raw point clouds, without suffering from limitations of topology, voxel resolution, density and surface ambiguity problem. In this work, we conduct experiments on four challenging benchmarks, where our method clearly outperforms all the existing methods including state-of-the-art NDF by a large margin. More importantly, our method shows superior generalization ability on unseen data across multiple datasets as detailed in the experiments section. Overall, our key contributions are summarized below:

- We propose a range-aware neural interpolation module with an unsigned distance fields function, which eliminates the surface ambiguity of naive trilinear interpolation in feature query to recover precise 3D scene geometry.
- We propose an implicit surface-oriented semantic segmentation module to infer semantic information for the implicit surface, benefiting from strong geometric priors provided by the joint optimization of reconstruction.
- Our extensive experiments demonstrate the potential for real-world applications with practicality on real-world data and showcase the superiority of our methods against all the existing methods and state-of-the-art approaches in accuracy and speed by a large margin on four datasets and presents very strong generalization capability across unseen data.

3.2 Method Overview

Given a sparse point cloud P of a 3D scene as input, which consists of N sparsely and non-uniformly sampled points from open surfaces with complex scene geometries, we aim to reconstruct the underlying continuous implicit surface geometries $\boldsymbol{S}_{geo}$ and the accurate semantic labels $\boldsymbol{S}_{sem}$ for the implicit surface. We formulate this problem as learning neural unsigned distance fields with semantic segmentation. This neural function $f(P, q)$ encodes the sparse point cloud $\boldsymbol{P}$ and takes an arbitrary query point $\boldsymbol{q}$ as input, and subsequently directly predicts the unsigned distance fields d_q between the query point q and its closest surface along with its semantic label s_q

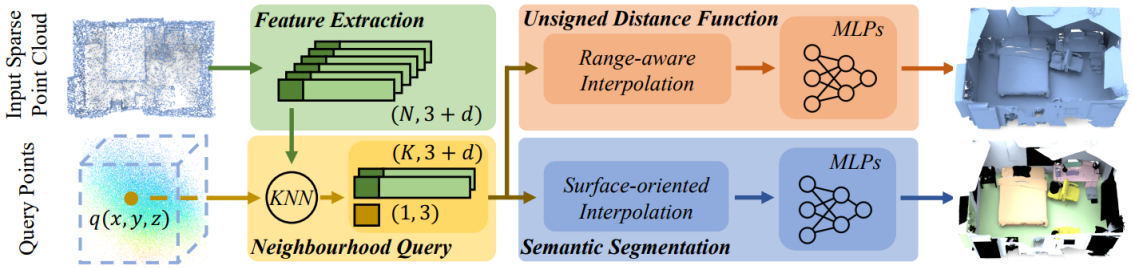


Figure 3.2: **RangeUDF overview**: Given an input point cloud, the feature extractor first extracts high-quality features for each point. This is subsequently followed by our novel range-aware unsigned distance function and surface-oriented segmentation module to learn precise geometry and semantics for each query point.

out of C classes. It is defined as below:

$$(d_q, s_q) = f(\boldsymbol{P}, \boldsymbol{q}); \quad q \in \mathbb{R}^3, d_q \in \mathbb{R}_0^+, s_q \in \mathbb{R}^C \tag{3.1}$$

As shown in Figure 3.2, our framework consists of four modules: 1) a per-point feature extractor in the top-left block, 2) the query point neighbourhood searching module in the bottom-left block, 3) the range-aware unsigned distance fields implicit function with neural interpolation module in the top-right block, and 4) the implicit surface-oriented semantic segmentation module in the bottom-right block. Details of these two modules are discussed below.

3.2.1 Point-wise Feature Encoding

Feature Extraction: This module extracts per-point features from a sparse input point cloud. As mentioned before, we adopt the large-scale point cloud friendly backbone RandLA-Net [7] for fast inference and our framework is not restricted to any specific backbone. Figure 3.3 shows the process of feature extraction: Given a raw point cloud of a scene with N points $\{p_1...p_n...p_N\}$ as on-surface points for implicit surface representation encoding, a 4-level encoder-decoder with skip connections is subsequently applied to learn a 32-d feature $\{F_1...F_n...F_N\}$ for each of the N input points hierarchically.

Neighbourhood Query: For the neighbourhood query operation, we imple-

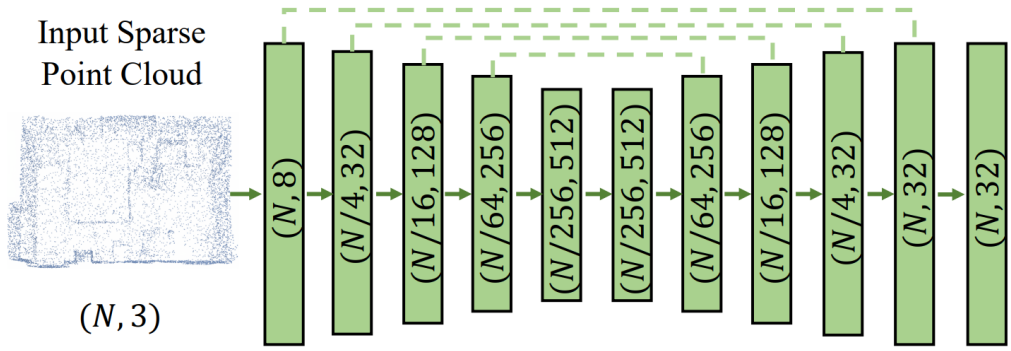


Figure 3.3: The detailed architecture of feature extractor. We modify the last layer of the decoder in RandLA-Net [7] to output a 32-dimensional feature vector for each surface point.

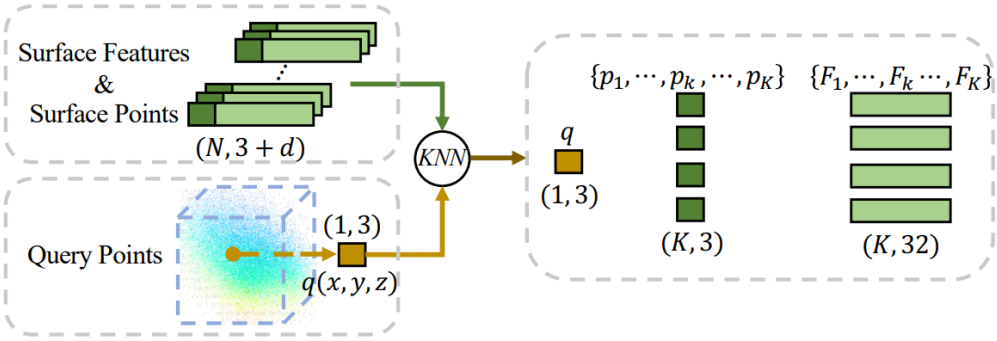


Figure 3.4: The details of neighbourhood query module.

ment an efficient $C++$ library for the KNN algorithm to collect K neighbours for each of the query points in batch based on the Euclidean distances, and other query methods such as spherical query [94] are also applicable. As presented in Figure 3.4, given a query point q, we first find the nearest K neighbours within the N input surface points. Subsequently, we retrieve the K neighbouring surface points $\{p_1...p_k...p_K\}$ of q and corresponding point features $\{F_1...F_k...F_K\}$. After collecting K points and corresponding point features for each query point q, we feed those into the range-aware neural unsigned distance fields module and surface-oriented semantic segmentation module to regress the unsigned distance and semantic classification for each query point q.

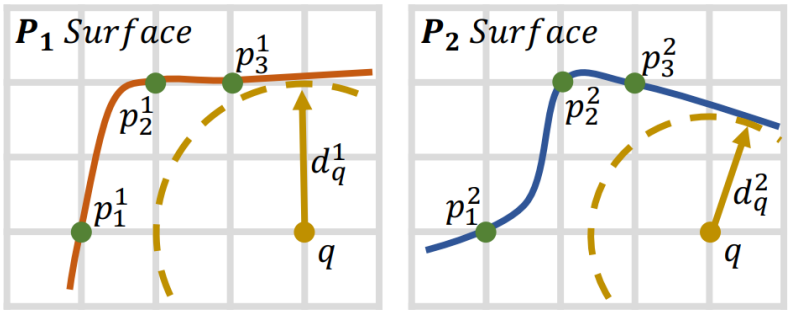


Figure 3.5: The ambiguity of simple trilinear interpolation.

3.2.2 Range-aware Unsigned Distance Function Module

Ambiguity of Trilinear Interpolation: Given the K neighbours and their features for a query point q, trilinear interpolation is widely used in existing works such as NDF [6] and ConvOcc [30] to learn a weighted feature for the query point q. However, such simple interpolation suffers from a distance ambiguity problem when the input point cloud is sparse or with complex structures during training and inference. For instance, as shown in Figure 3.5, given two different surfaces P_1 and P_2, for the same query point q, it is very likely that the two sets of queried features of $\{p_1^1, p_2^1, p_3^1\}$ on surface P_1 and $\{p_1^2, p_2^2, p_3^2\}$ on surface P_2 will be very similar or identical.

However, due to the voxel representation, sparsity, and complexity of point clouds, the two underlying surfaces of P_1 and P_2 can be significantly different while having the same vertex position, which means that the unsigned distance fields d_q^1 and d_q^2 from q to P_1 and P_2 could be completely different as shown in Figure 3.5. In this case, naive trilinear interpolation will cause ambiguity in representing the distance fields using these identical feature vectors. During training, such ambiguity will confuse the network and tend to predict a *mean* unsigned distance fields between the ground truth d_q^1 and d_q^2. This will essentially result in over-smooth surfaces during inference.

Range-aware Neural Interpolation: As shown in Figure 3.2 and Figure 3.6, to explicitly avoid the ambiguity demonstrated above, we leverage a simple yet effective range-aware neural interpolation module. As shown in Figure 3.6, given a query point q, we first find its K nearest neighbours $\{p_1 ... p_k ... p_K\}$ and their

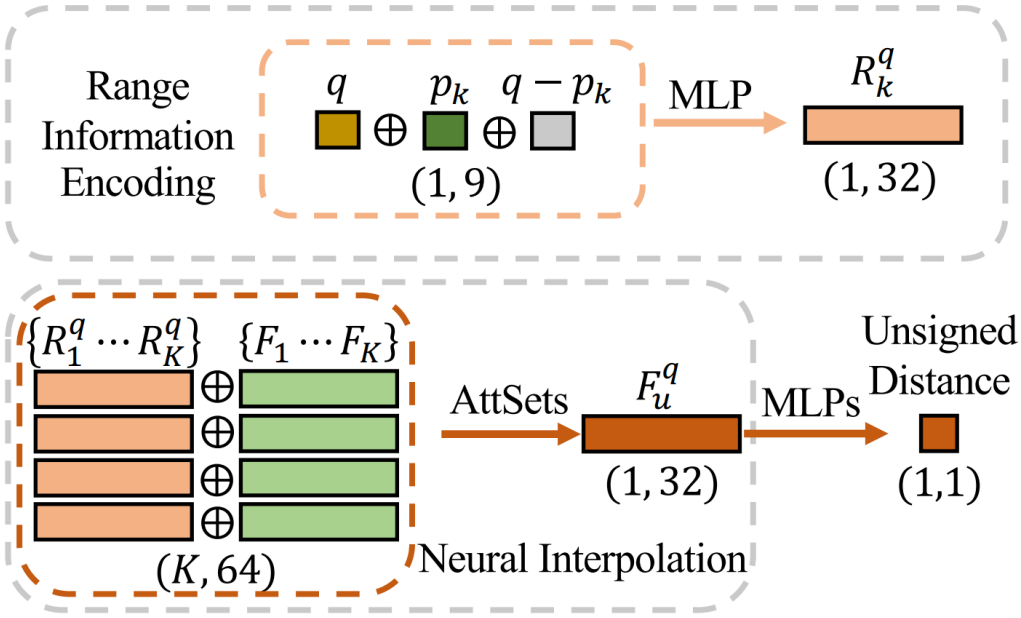


Figure 3.6: The details of the range-aware unsigned distance function.

corresponding feature vectors $\{F_1...F_k...F_K\}$. Subsequently, our range-aware neural interpolation module explicitly takes relative distances and absolute positions of all neighbouring points into consideration. Specifically, we encode the range and distance information for each neighbour as follows:

$$\boldsymbol{R}_k^q = \boldsymbol{MLP}((q - p_k) \oplus q \oplus p_k) \tag{3.2}$$

where q and p_k are the (x, y, z) positions of points q and p, and $\oplus$ is the concatenation operator. As shown in the top block in Figure 3.6, the input of MLP is a concatenated 9-d position vector and the output is a 32-d range vector R_k^q. To gain scale invariance, we normalize all the input point clouds into a cube with the size range of $[-0.5, 0.5]$ along (x, y, z) axes following prior art [6]. As illustrated in Figure 3.7, for a query point q, if the surface patches that contain queried nearest neighbours of the point cloud P_1 and P_2 are similar to each other but having different distance fields and position shift, the term of relative position $(q - p_k)$ can explicitly guide the network to be aware of the difference between the unsigned distance d_q^1 and d_q^2 and to learn a distinctive feature vector. Section 3.4.6 provides additional supporting experimental evidence to support this thinking.

Traditional trilinear interpolation simply computes a set of weights $\{w_1^q...w_k^q\}$ by

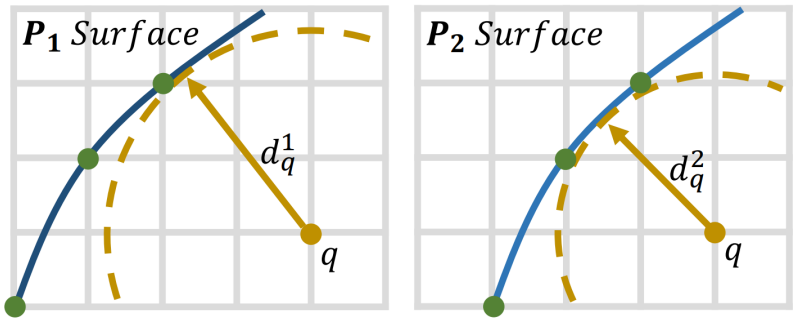


Figure 3.7: The importance of relative distance.

computing Euclidean distance between the query point q and their K neighbouring surface points $\{p_1...p_k...p_K\}$. Whilst our method learns a set of informative vectors $\{R_1^q...R_k^q...R_K^q\}$, which are explicitly aware of the distance and range between them, eliminating the distance ambiguity of trilinear interpolation. To interpolate a single feature vector F_u^q for the query point q, we concatenate the range vectors with point features followed by an attention pooling layer. Specifically, our neural interpolation module is defined as follows:

$$F_u^q = \mathcal{A}([R_1^q \oplus F_1]...[R_k^q \oplus F_k]...[R_K^q \oplus F_K]) \tag{3.3}$$

where $\mathcal{A}$ is the simple attention block AttSets [95], although Transformer [96] could possibly yield better results. As shown in the bottom block in Figure 3.6, the input of AttSets are K concatenated 64-d vectors and the output is a 32-d feature vector F_u^q.

Unsigned Distance Regression: Finally, we feed the feature vector F_u^q of query point q into 4 MLPs to regress the unsigned distance fields d_q, where the dimensions of the MLPs are $(512 \rightarrow 32 \rightarrow 32 \rightarrow 1)$. A LeakyReLU ($s$=0.2) function is integrated into the first 3 layers. The last MLP is subsequently followed by a *ReLU* activation function, enabling the unsigned distance value to be equal or greater than 0, because UDFs should always be equal or greater than 0.

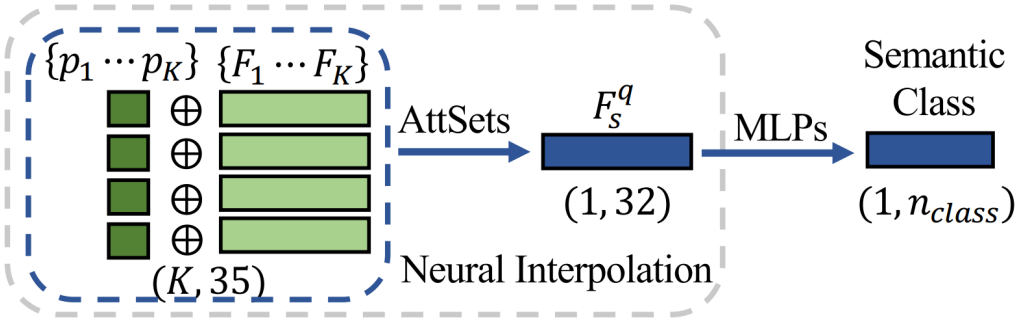


Figure 3.8: The details of surface-oriented semantic segmentation.

3.2.3 Surface-oriented Semantic Segmentation Module

Absence of Semantics for Implicit Surfaces: The key difference between learning unsigned distance fields and learning valid semantic classes for continuous surfaces is that for the query points q located in an empty space, there will not be meaningful and valid semantic labels for supervision. Naturally, the semantic labels will only present at a valid surface patch and only the vertices and faces on the surface can have semantic labels. However, our surface representation is composed of an implicit underlying function. As a result, the main problem for us is how to learn meaningful semantic labels for implicit surfaces. A naive strategy is to only supervise the semantic labels for the points on the surface (on-surface points) and separately supervise unsigned distance fields for both on-surface and off-surface points, which are randomly sampled points in testing or boundary sampled points in training. However, this kind of strategy will inevitably cause imbalance and ineffective optimization of the two branches as shown in the experiments section.

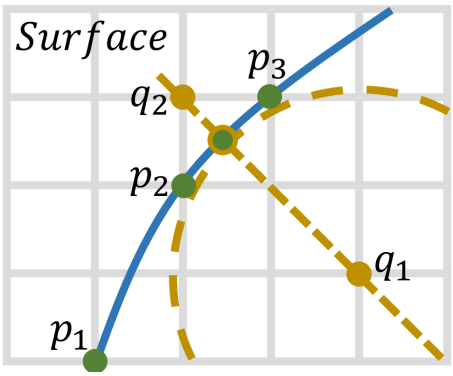


Figure 3.9: Eliminating the absolute position of the query point.

Implicit Surface Oriented Semantic Segmentation: To learn meaningful and accurate semantic information for implicit surfaces, we introduce a surface-oriented semantic segmentation module as shown in Figure 3.8. Specifically, given a query point q and its K nearest neighbours $\{p_1...p_k...p_K\}$ along with their corresponding feature vectors $\{F_1...F_k...F_K\}$, this module will only leverage the information of the neighbours to predict the semantic label for the query point q and ignore the absolute position information of the **query point** $\boldsymbol{q}$. Formally, the semantic label $\boldsymbol{s_q}$ for the query point q is defined as:

$$\begin{aligned} F_s^q &= \mathcal{A}([p_1 \oplus F_1]...[p_k \oplus F_k]...[p_K \oplus F_K]), \\ s_q &= MLPs(F_s^q) \end{aligned} \tag{3.4}$$

where $\mathcal{A}$ is also the attention function Attsets [95] to aggregate the K feature vectors. Specifically, the input is K concatenated 35-d vectors and the output is a 32-d logit as semantic feature vector F_s^q. Subsequently, we regress the semantic class for the query point q from its semantic feature vector F_s^q through 3 MLPs. The output dimensions in our experiments are $(64 \rightarrow 32 \rightarrow C)$, where C is the number of classes. Following the same thoughts as in the above interpolation module, a LeakyReLU ($slope$=0.2) is integrated into the first two layers of MLP.

As shown in Figure 3.9, the key aim of the formulation above is to learn the semantic labels for the nearest surface patch $\boldsymbol{P}$ composed by the K nearest neighbours $\{p_1...p_k...p_K\}$ instead of the discrete query point q. For instance, given an existing surface patch P formed by K nearest neighbours $\{p_1...p_k...p_K\}$, for all the neighbouring query points such as p_1 and p_2 with different unsigned distance fields, this module will guide the network to learn a consistent semantic class over the implicit surface patch near the underlying discrete surface patch for all the neighbouring query points. In this manner, we no longer suffer from the sensitivity and ambiguity brought by absolute distances.

3.2.4 Loss function

For training RangeUDF, our optimization aim is to let our method $f_P(q)$ gain accurate unsigned distance fields $\boldsymbol{S}_{geo}$ and semantic classes $\boldsymbol{S}_{sem}$ for implicit surfaces.

Surface Reconstruction Loss: For an input sparse surface P, To supervise our reconstruction branch, we adopt an $\mathcal{L}_1$ loss to supervise the UDFs:

$$\begin{aligned} \hat{d_q} &= UDF(P, q), \\ \mathcal{L}_{rec} &= \sum_q ||min(\hat{d_q}, \epsilon) - min(d_q, \epsilon)||_1^1 \end{aligned} \tag{3.5}$$

where $\hat{d_q}$ is the ground truth unsigned distance fields for the query point q and d_q are the predicted unsigned distance fields and ϵ is the threshold to clamp the distance fields that are too far away from the surfaces to strengthen the ability to represent the vicinity of the scene surface.

Implicit Surface Segmentation Loss: To gain semantic information for the implicit surface reconstructed by our neural implicit function, we adopt CrossEntropy Loss:

$$\mathcal{L}_{sem} = CrossEntropy(\sigma(S_{sem}), \hat{S_{sem}}) \tag{3.6}$$

where $\hat{S_{sem}}$ is the ground truth semantic label and $\sigma(S_{sem})$ is the predicted semantic probabilistic logits after softmax σ.

Total Loss: To avoid manually adjusting the weights between the reconstruction branch and the segmentation branch, we adopt an uncertainty loss proposed in [97]. As a result, our final loss can be represented as:

$$\mathcal{L}_{total} = e^{-\gamma}\mathcal{L}_{rec} + \gamma + e^{-\beta}\mathcal{L}_{sem} + \beta \tag{3.7}$$

where γ and β are initialized as 0 and then learned to balance two branches.

3.3 Implementation

We implement our method and conduct all of the experiments with Pytorch [98]. The experiments in this chapter is all conducted on Intel(R) Xeon(R) E5-2698 v4

@ 2.20GHz CPU and an NVIDIA Tesla V100 GPU.

Training: Our method is trained in an end-to-end manner without any pre-training or post-processing with a batch size of 4 on all of the datasets. The number of nearest neighbours is set to $K = 4$. We use Adam optimizer for training with a learning rate of 10^{-3} and default settings and parameters. For each point cloud, we uniformly sample 10k points as the input for implicit surface encoding. For the implicit surface generation, we first randomly sample 50k points in the empty space within a cube with a size of [-0.75, 0.75] along (x, y, z) axes and then feed them into the neural interpolation module for each iteration. It is worth noting that our method can not only infer much faster than the previous state-of-the-art approaches such as NDF [6] because of efficient point-wise encoding strategy, but we can also converge $5 \sim 10$ times faster than NDF. For the reproduction of our results in the paper and this chapter, 20 hours are enough for all of the datasets. In addition, our performance also outperforms their method by a remarkable margin on scenes.

Explicit Semantic Surfaces Extraction: During inference, we adapt from the algorithm proposed in NDF [6] to fit in our case. Specifically, the $f(q)$ of our encoded implicit neural field is used as an approximation of UDF. As a result, for a query point q, its projection $\hat{q}$ on the underlying surface P can be recovered by:

$$\hat{q} = q - f(q) \cdot \nabla_q f(q), \quad \hat{q} \in P, \quad \forall q \in \mathbb{R}^3 / L \tag{3.8}$$

where ∇_q is the gradient of q related to the implicit function and $\hat{q} \in \mathbb{R}^3$ is the final projected position of query point q, and L is the cut locus [99]. As an approximation of UDF, the direction of the negative gradient of q is the shortest path from q pointing to the underlying surface.

The full process is described as in Alg. 1. First, after encoding the implicit surface in $f(q)$, we randomly sample 200K points in the cube with a size of [-0.75, 0.75] along (x, y, z) axes. Subsequently, we project points q with valid unsigned distance fields $f(q) < \epsilon\}$ by $num\_steps = 7$ times as an initialization P_0, where $\epsilon = 10cm$. After that, to get a dense point cloud, we first do a Gaussian sampling with a variance of $\epsilon/3$ from the sparse P_0 as P_{dense}, and project them again for $num\_steps = 7$

Algorithm 1 Dense point cloud generation

Input: Implicit function $f(q)$ and P_0 (m points uniformly sampled in the cube)
Output: P_{out} (generated dense point cloud)

$P_0 \leftarrow \{q \in P_0 | f(q) < \epsilon\}$
for $i = 1$ `to` num_steps **do**
    $\hat{q} = q - f(q) \cdot \frac{\nabla_q f(q)}{||\nabla_q f(q)||}, \quad \forall q \in P_0$
end for ▷ The end of initialization
$P_{dense} \leftarrow \{q + d | q \in P_0, d \sim \mathcal{N}(0, \epsilon/3)$
while $N < max\_num\_points$ **do** ▷ N is the number of points in P_{out}
    for $i = 1$ `to` num_steps **do**
        $\hat{q} = q - f(q) \cdot \frac{\nabla_q f(q)}{||\nabla_q f(q)||}, \quad \forall q \in P_{dense}$
        $P_{out} \leftarrow \{\hat{q} \in P_{dense} | f(\hat{q}) < \epsilon\}$ ▷ Concatenate $\hat{q}$ to P_{out}
    end for
end while
return P_{out}

times to gain better surface reconstruction before adding them into the final results P_{out}, because the $f(q)$ is only approximation to UDF and there will be inaccuracies. Subsequently, the implicit semantic labels are also retrieved for $\hat{q}$ before adding into P_{out} after the process ends. The process ends when the point number exceeds $max\_num\_points = 1600k$. Finally, off-the-shelf meshing algorithms could be applied for mesh extraction, such as [85, 100]. In our experiments and reproduction of NDF, we use Marching Cubes [85] to extract meshes with semantics for evaluation of all the methods including NDF. Although the Ball-Pivoting algorithm [100] can also produce similar results, it is slow and inefficient.

Simple meshing strategy: To better visualize the reconstruction quality of our method, we first predict the unsigned distance field value for each voxel that lies in the volume with a resolution of 256^3. This process will take 0.95 seconds for our method and around 15 seconds for NDF [6]. Subsequently, we feed the volume into the Marching Cubes algorithm with $level = 0.003$ and $spacing = [1.0/255] * 3$ to generate the mesh.

3.4 Experiments

We evaluate our RangeUDF in two aspects: 3D surface reconstruction and semantic segmentation. We conduct extensive experiments on four datasets: Synthetic Rooms

[30], ScanNet [5], 2D-3D-S [12] and SceneNN [11]. Moreover, we jointly evaluate the performance on three real-world datasets (except the Synthetic Rooms dataset, which is a simple dataset proposed in [30] without semantic label), and we investigate how our reconstruction branch and semantic branch complement and benefit each other.

For a fair comparison with ConvOcc [30] and SA-ConvOnet [89], we use their official pre-trained models and codes. However, as NDF [6] only conducts object-level reconstruction on ShapeNet Cars dataset [20], and there is no scene-level reconstruction in their official implementation, we carefully discuss with the authors of the paper and adapted it following the instructions in their paper [101] and we also improved the performance of NDF.

3.4.1 Datasets

For all four datasets, we use the official training/validation/testing splits. Note that, only the Synthetic Rooms dataset consists of closed 3D surfaces, while the other three are real-world datasets with complex topology and noisy open surfaces.

ScanNet [5] contains 2.5 million views in 1513 real-world rooms captured by a scalable RGB-D capture system. The annotations include camera poses, surface meshes and panoptic segmentations with both semantic and instance information. There are 20 semantic classes available in this dataset. We follow the official split to use 1,201 rooms for training and 312 for evaluation. For surface reconstruction, we directly sample sparse points as on-surface points from the real-world non-watertight meshes provided in the test split.

2D-3D-S [12] contains 6 large-scale indoor point clouds with 271 rooms in total (From Area-1 $\sim$ Area-6, there are 44, 40, 23, 29, 49, 67, 48 rooms separately). For each of the rooms, a semantically annotated non-watertight mesh is captured by Matterport sensors. There are 13 semantic classes used for this dataset. Note that, we ignore Area-5 in this dataset because the ground mesh of this area is unevenly broken into 2 parts with many false points and artefacts. Therefore we forego this area since the official repository does not provide information for recovering the whole mesh and refining the quality to be as good as other areas. As a result,

we use Area-1 $\sim$ Area-4 for training and Area-6 for evaluation following the most commonly used protocol.

Synthetic Rooms [30] is a simple synthetic dataset proposed in ConvOcc [30], which contains 5000 scenes. Each scene consists of multiple objects retrieved from the ShapeNet database [20], including common indoor furniture such as chairs, sofas, lamps, cabinets and tables. We follow the official split proposed in the paper and test on the whole test set in our experiments. This dataset is only used for experiments of generalization of reconstruction since there are no semantic labels.

SceneNN [11] is an RGB-D scene dataset that contains 76 indoor scenes captured at various places. For each scene, the reconstructed triangle mesh is provided, and there are 11 semantic classes for this dataset. We follow the official split in our experiments to use 56 scenes for training and 20 scenes for evaluation.

Data Preparation: For the input point cloud, we first follow prior arts [6, 30] to normalize the ground truth mesh into a unit cube with a size of [-0.5, 0.5] along (x, y, z) axes. Subsequently, to generate on-surface points for implicit surface encoding, we sample 10k points from the normalized mesh for a fair comparison. To generate off-surface points for feature query and supervision, we first sample 100k points from the vicinity of the ground truth surface by Gaussian sampling with $\sigma = 0.08, 0.02, 0.003$ and mix them by a ratio of $1\%, 49\%$ and 50% separately following NDF [6]. Subsequently, we further randomly sub-sample 50k points for training from the Gaussian sampled points. For each point on the surface, we assign it to have the same semantic label as its nearest face of the ground truth surface. Finally, with the (x, y, z) positions of the sparse raw point clouds and all query points with their unsigned distance fields and semantic labels, we train our RangeUDF in an end-to-end manner.

3.4.2 Evaluation Metrics

To evaluate the reconstruction quality, we follow prior arts [6, 30] to compare the ground truth surface and the point clouds sampled from predicted implicit surfaces and use the popular Chamfer-L1 Distance, Chamfer-L2 Distance and F-score with a distance threshold of (FS-δ, FS-2δ, FS-4δ, $\delta = 0.005$). For semantic segmentation,

we adopt the commonly used Mean Intersection Over Union (mIoU) and Overall Accuracy (OA). For simplicity, we use (↑) and (↓) to represent that the metric is better when the value becomes larger or smaller.

Chamfer-L1 Distance (↓): Chamfer-L1 Distance [34] is commonly used for evaluating the shapes due to its simplicity. It is defined as the mean of an accuracy and a completeness metric, where the accuracy is the mean distance from the predicted surface points to their corresponding nearest neighbours on the ground truth surface. The completeness is similar but in an opposite direction. Specifically, it can be computed as:

$$CD(P_G, P_R) = \sum_{g \in P_G} \min_{r \in P_R} ||g - r||_2^2 + \sum_{r \in P_R} \min_{g \in P_G} ||r - g||_2^2 \tag{3.9}$$

where (g, r) is points from the ground truth surface and predicted reconstruction $(P_G, P_R) \in \mathbb{R}^3$ and the two terms are called completeness and accuracy. Note that d_{CD} is technically not a valid distance function since the triangle inequality does not hold in this metric [34], but it is nevertheless be termed as a pseudo distance function due to its non-negative property. To find the nearest neighbours between the two sets of points, we use an efficient KD-Tree to estimate the corresponding distances. Although it is simple, it produces high-quality and reasonable results in practice. Following prior arts [6, 30, 72] to report the mean value over the 100k sampled points with a scaling coefficient $\times 10^{-2}$ for better readability.

Chamfer-L2 Distance (↓): Having the Chamfer-L1 Distance above, it is natural to extend to the L2 form by using the squared distance before we compute the mean value of the final results. For better readability with a decimal number with a lot of proceeding zeros, we use $\times 10^{-4}$ to scale up the value in the table.

F-score (↑): We also evaluate F-score [102] following prior arts [30, 103]. This is further defined by the accuracy and completeness mentioned above. F-score ($F(\tau)$) is formed by two parts: Recall $R(\tau)$ and Precision $P(\tau)$, where τ is the distance

thresholds. Specifically, they can be represented as:

$$d_{g\to r} = \min_{r\in P_R} ||g - r||, \quad d_{r\to g} = \min_{g\in P_G} ||r - g||_2^2, \tag{3.10}$$

$$R(\tau) = (\frac{100}{|G|}) \sum_{g\in G} [d_{g\to r} < \tau], \tag{3.11}$$

$$P(\tau) = (\frac{100}{|R|}) \sum_{r\in R} [d_{r\to g} < \tau], \tag{3.12}$$

$$F(\tau) = \frac{2R(\tau)P(\tau)}{R(\tau) + P(\tau)} \tag{3.13}$$

where P_G and P_R is ground truth and our reconstruction and $[\cdot]$ is the Iverson bracket (1 if the condition in brackets is satisfied else 0) and τ is the distance threshold. The F-score at a given threshold d is the harmonic mean of precision and recall.

mIoU (↑): Mean Intersection-Over-Union (mIoU) is the one of the most commonly used metric for evaluating the quality of segmentation. Similar to image domain, mIoU is calculated as:

$$mIoU = \frac{1}{C} \sum_{i}^{C} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{3.14}$$

where TP_i, FP_i, FN_i are the true positive, false positive and false negative samples of class i.

OA (↑): Overall accuracy is also used to evaluate the semantic segmentation quality of our method, which can be represented as:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.15}$$

where a true positive represents a point that is correctly predicted to be the same as the ground truth class and this metric is evaluated as the ratio of correct predictions.

3.4.3 3D Scene Surface Reconstruction

To fully showcase and evaluate the reconstruction ability of RangeUDF, we conduct two sets of experiments: 1) Reconstruction on the four benchmark datasets separately and 2) generalization tests on unseen datasets. In the reconstruction ex-

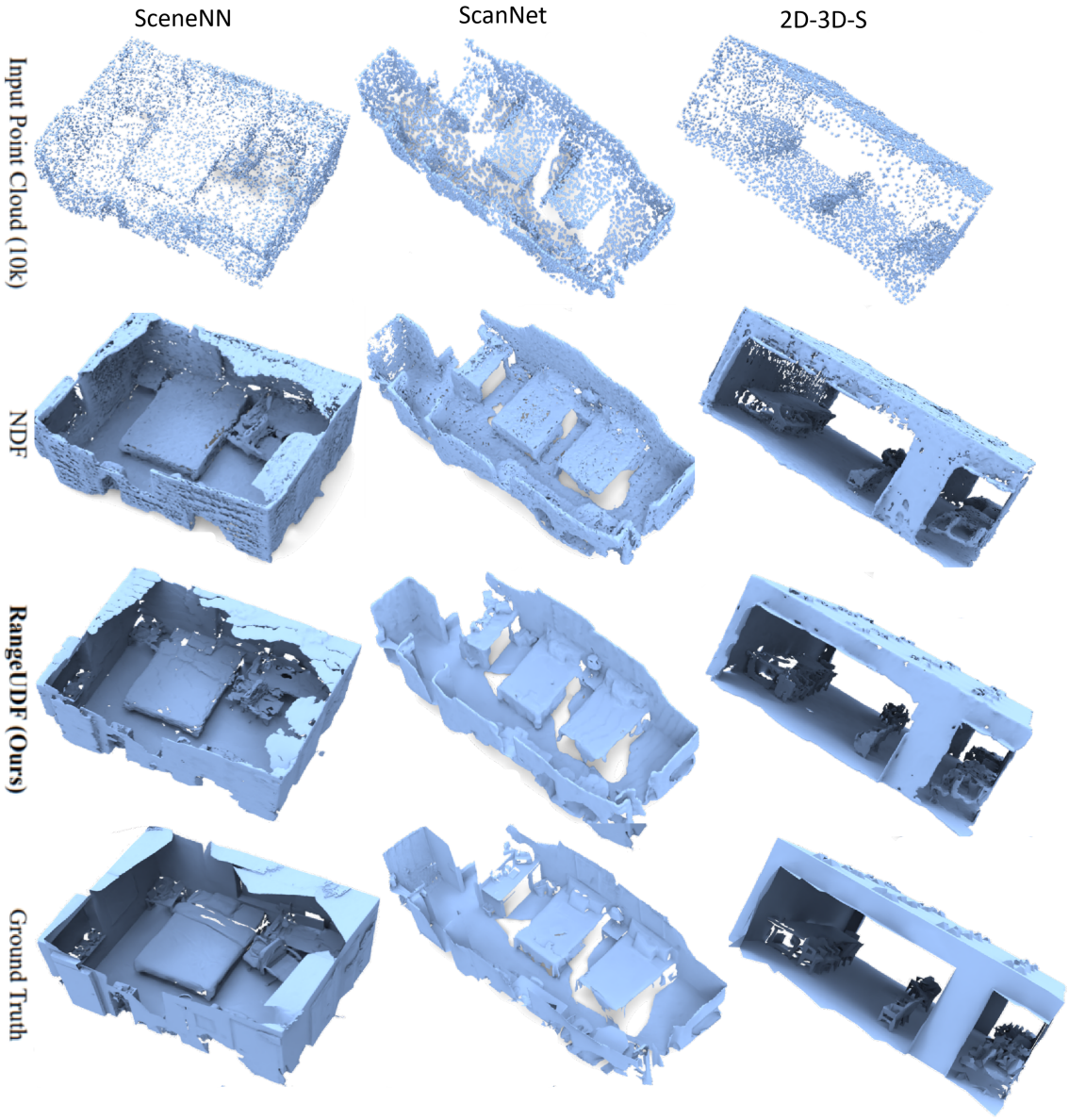


Figure 3.10: **Qualitative comparison of surface reconstruction** on three challenging real-world dataset: ScanNet, SceneNN and 2D-3D-S. For a fair comparison with NDF, we use the same level value to generate the surface meshes with the Marching Cubes algorithm.

periments, we follow prior arts [6, 30, 72] to sample 10k points for surface encoding and 100k points for training (50k random sub-samples are used for each iteration). For evaluation, we first generate a dense surface by the Algorithm 1. Subsequently, we follow the same protocol to sample 100k points from the predicted surface and ground truth surface to compute the metrics. It is worth noting that it only takes 9.8 seconds on average for us to generate the whole scene with semantic labels while NDF [6] needs approximately more than 1 minute to recover only the geometric information depending on the actual size of the scene.

	SceneNN				ScanNet				2D-3D-S			
Metrics	CD- L_1	CD- L_2	FS- δ	FS-2 δ	CD- L_1	CD- L_2	FS- δ	FS-2 δ	CD- L_1	CD- L_2	FS- δ	FS-2 δ
NDF	0.460	0.248	0.726	0.927	0.385	0.214	0.800	0.964	0.418	0.523	0.762	0.969
Ours	**0.327**	**0.169**	**0.834**	**0.977**	**0.286**	**0.125**	**0.884**	**0.988**	**0.327**	**0.194**	**0.845**	**0.977**

Table 3.1: **Quantitative comparison** of our RangeUDF and NDF on three real-world datasets: SceneNN, ScanNet, and 2D-3D-S.

Results on Real-World Benchmarks: As mentioned above, we conduct independent experiments on the four datasets separately and all the methods are trained and tested on the corresponding single dataset following the official split. Table 3.1 compares our method with NDF on the three challenging real-world datasets: ScanNet [5], SceneNN [11] and 2D-3D-S [12]. It can be clearly seen that our method outperforms NDF [101] by a large margin on the three datasets and under all the metrics. Figure 3.10 presents a qualitative comparison with NDF on three datasets, where it can be seen that our method successfully recovers the fine-grained scene geometry and high-quality continuous surfaces, but NDF still retains many artefacts and holes on the surfaces. This further demonstrates the superiority of our simple range-aware neural interpolation module to recover complex 3D scenes with arbitrary topologies from real-world data.

Results on Synthetic Rooms Dataset: Except NDF [6], other prior arts fail to represent **open surfaces** due to the lack of powerful representation and they are limited to manually closed watertight mesh. As a result, we can only compare with these solid methods on Synthetic Rooms Dataset [30]. Due to its perfectly closed 3D surfaces, it is possible for all types of implicit representation methods and explicit methods. As shown in Table 3.2, our method significantly outperforms all the other

Methods	CD- L_1	CD- L_2	FS- δ	FS-2δ	FS-4δ
SPSR	2.083	–	–	0.762	0.812
Trimmed SPSR	0.690	–	–	0.892	–
PointConv	1.650	–	–	0.790	–
OccNet	2.030	–	–	0.541	–
SAL	2.720	–	–	0.405	0.598
IGR	1.923	–	–	0.740	0.812
LIG	1.953	–	–	0.625	0.710
ConvOcc	0.420	0.538	<u>0.778</u>	<u>0.964</u>	0.983
NDF	<u>0.408</u>	<u>0.301</u>	<u>0.713</u>	<u>0.952</u>	<u>0.998</u>
SA-CONet	0.496	0.686	0.747	0.936	0.986
RangeUDF	**0.348**	**0.179**	**0.803**	**0.978**	**0.999**

Table 3.2: Quantitative comparison of our RangeUDF and other prior arts on Synthetic Rooms dataset. The underline and bold represent the second best and the best separately.

methods by a large margin under all the metrics. This further demonstrates the advantages of our powerful representation to represent all kinds of topologies for both real-world data and synthetic data.

3.4.4 Generalization Experiments

It can be seen from the above experiments that we are already clearly the state-of-the-art approach with superiority against all the other methods on both the Synthetic Room dataset and the other three challenging real-world datasets under all the evaluation metrics. In this part of the experiments, we demonstrate the capability of our method to generalize across different unseen datasets. Specifically, we train the methods on one of the four datasets and subsequently test it on the other three datasets. For testing the ability to bridge the domain gap between synthetic data and real-world data, we further compare our method with ConvOcc [30], NDF [6] and SA-CONet [89]. For real-world data generalization, we can only compare with NDF since other methods cannot represent open surfaces.

Quantitatively, it can be seen from Table 3.3 that our RangeUDF clearly surpasses all the other methods in all the evaluation protocols. Our method is the only one that can perform consistently across all the unseen datasets. Especially, we can

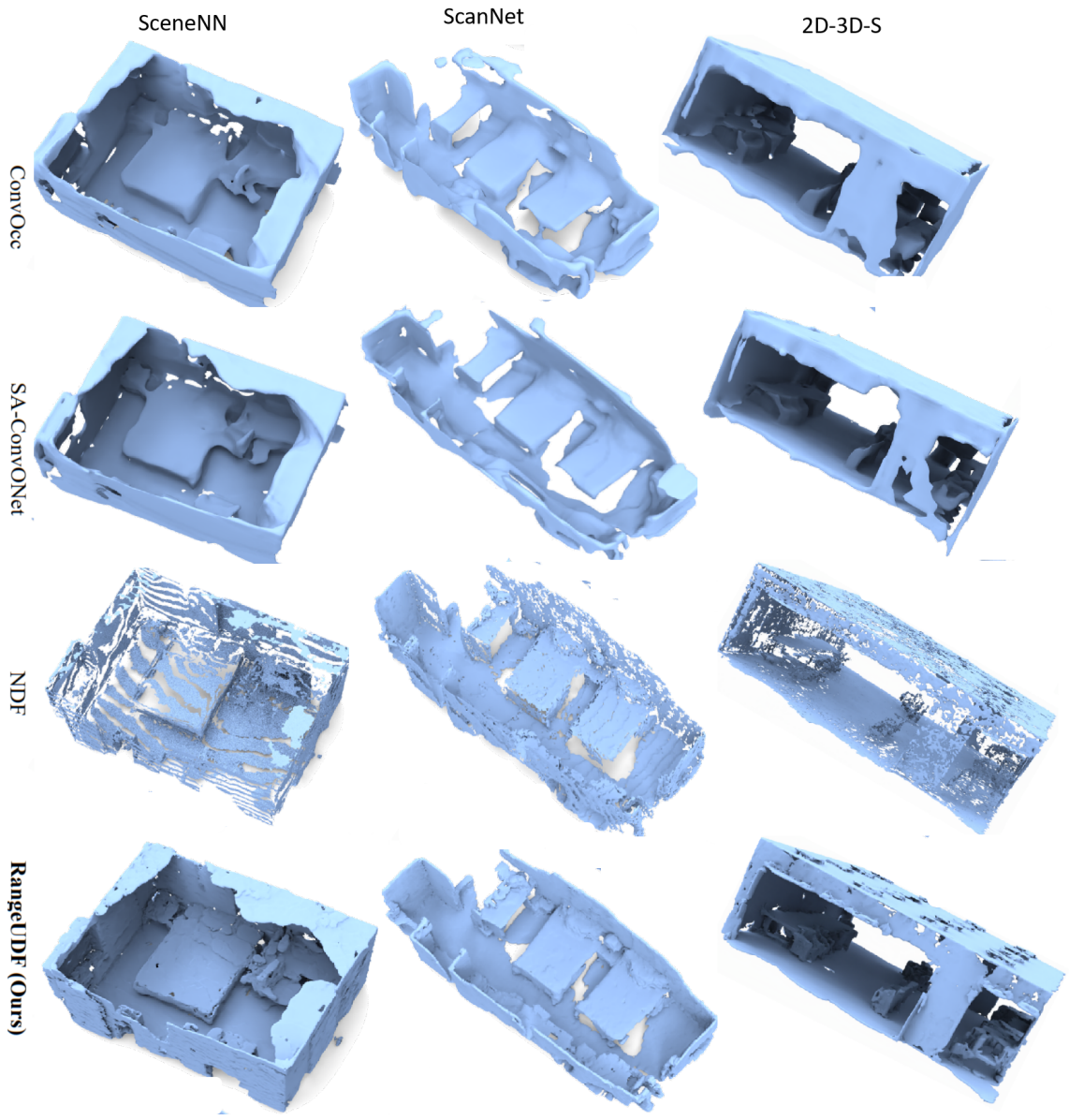


Figure 3.11: **Qualitative comparison of generalization** on three challenging real-world dataset: ScanNet, SceneNN and 2D-3D-S. For a fair comparison with NDF, we use the same level value to generate the surface meshes with the Marching Cubes algorithm.

perform stably under the setting of cross-domain generalization.

Trained on	CD-L_1	CD- L_2	FS-δ	FS-2δ	CD-L_1	CD-L_2	FS-δ	FS-2δ	CD-L_1	CD-L_2	FS-δ	FS-2δ
Synthetic	Tested on SceneNN				Tested on ScanNet				Tested on 2D-3D-S			
ConvOcc	0.816	1.733	0.421	0.786	0.845	1.902	0.397	0.778	0.960	2.433	0.323	0.884
NDF	0.455	0.286	0.649	0.962	0.452	0.281	0.648	0.960	0.468	0.286	0.609	0.969
SA-Conv	0.744	1.223	0.393	0.836	0.776	1.662	0.346	0.833	0.874	1.983	0.303	0.811
Ours	**0.332**	**0.176**	**0.827**	**0.975**	**0.303**	**0.139**	**0.864**	**0.986**	**0.327**	**0.160**	**0.838**	**0.981**
SceneNN	Tested on Synthetic Rooms				Tested on ScanNet				Tested on 2D-3D-S			
NDF	0.569	0.458	0.404	0.868	0.462	0.389	0.707	0.928	0.688	1.712	0.662	0.858
Ours	**0.474**	**0.407**	**0.627**	**0.904**	**0.285**	**0.127**	**0.880**	**0.989**	**0.340**	**0.190**	**0.826**	**0.977**
ScanNet	Tested on Synthetic Rooms				Tested on SceneNN				Tested on 2D-3D-S			
NDF	0.568	0.431	0.401	0.881	0.425	0.273	0.730	0.948	0.442	0.284	0.698	0.948
Ours	**0.481**	**0.489**	**0.607**	**0.915**	**0.324**	**0.166**	**0.837**	**0.978**	**0.329**	**0.164**	**0.834**	**0.980**
2D-3D-S	Tested on Synthetic Rooms				Tested on SceneNN				Tested on ScanNet			
NDF	0.527	1.799	0.645	0.972	0.382	0.217	0.780	0.970	0.378	0.205	0.787	0.972
Ours	**0.432**	**0.310**	**0.654**	**0.929**	**0.314**	**0.161**	**0.845**	**0.978**	**0.272**	**0.112**	**0.898**	**0.991**

Table 3.3: Quantitative comparison of our RangeUDF and prior arts on generalization test across unseen datasets.

Qualitatively, in Figure 3.11, we provide the generalization results on three real-world datasets by training on the Synthetic Room dataset only. It is shown that our method significantly outperforms other methods and we are able to recover clearly finer surface details. Interestingly, ConvOcc [30], NDF [6] and SA-CONet [89] all adopt naive trilinear interpolation to query features for each query point. As a result, they can only produce over-smooth surfaces without details. This further demonstrates the superiority of our proposed range-aware neural interpolation module. Our method is the only one that can produce consistently finer detailed surfaces across all the unseen datasets even in cross-domain settings.

3.4.5 Joint Implicit Segmentation and Reconstruction

In addition to superior 3D surface reconstruction, our method is also the only one that can simultaneously recover semantic information for the implicit surface while all the other implicit representations can not. It is known that there are many existing prior arts that can perform very promising semantic segmentation for point clouds such as [7, 87, 94]. However, they are essentially working on a vastly different scenario compared to our case and it is not fair to directly compare with them on the available semantic segmentation benchmarks. Note that, our method does not aim

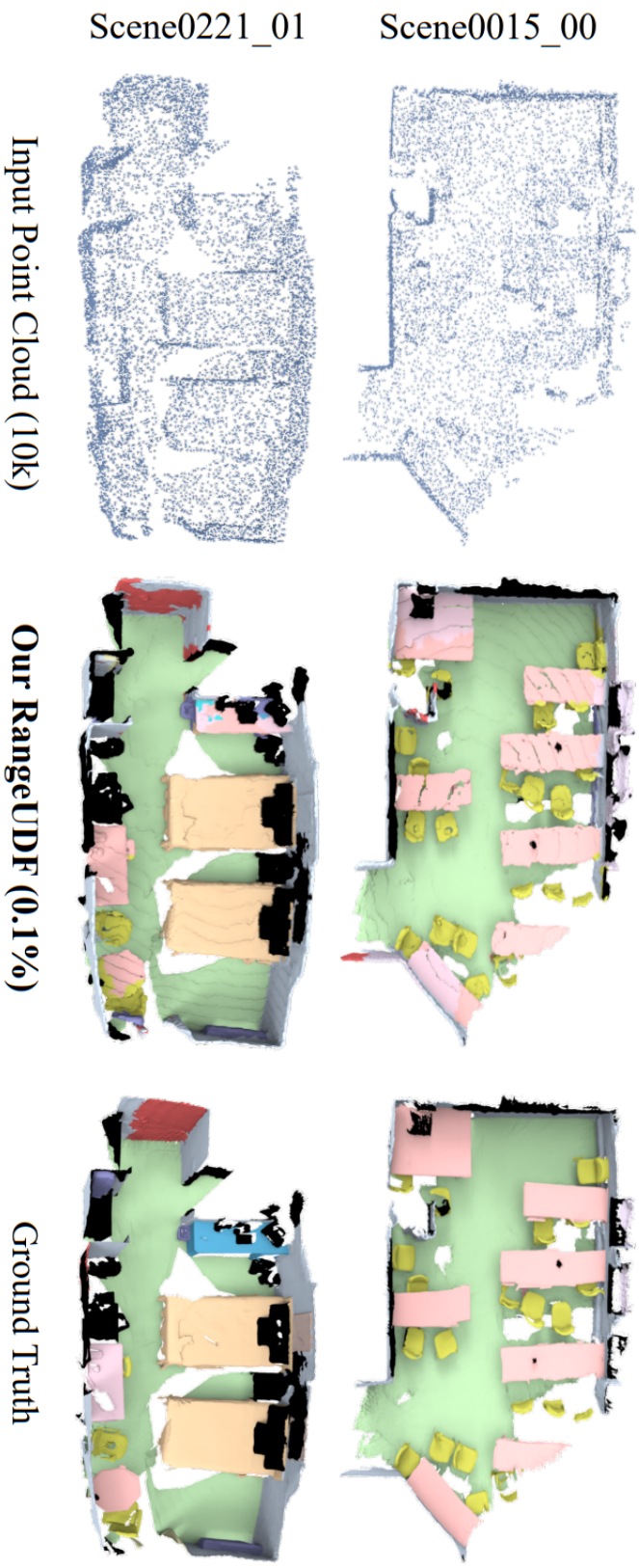


Figure 3.12: **Qualitative results of joint reconstruction-semantic optimization** on ScanNet [5] dataset given very limited supervision.

to achieve the best performance for a presented discrete point cloud, instead, our segmentation is performed effectively for continuous underlying surfaces represented by our neural implicit representation.

In this section, we jointly evaluate our network on three real-world datasets with semantic annotations: ScanNet [5], SceneNN [11] and 2D-3D-S [12] and conduct two groups of experiments to analyze 1) how the semantic information is learnt for implicit surfaces effectively and 2) how reconstruction branch and semantic branch benefit each other.

1) Does semantic branch degrade surface reconstruction?

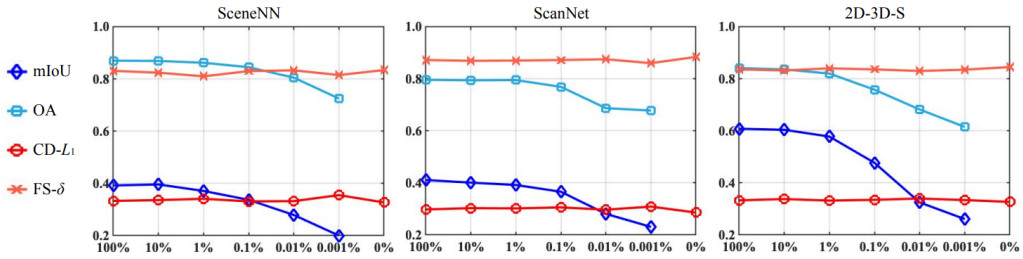


Figure 3.13: **Quantitative results of joint reconstruction-semantic optimization** on the three real-world datasets, given different amounts of supervision signals.

In this group of experiments, we jointly optimize the reconstruction branch with the range-aware neural interpolation module and the implicit surface-oriented segmentation module. Specifically, we supervise the semantic branch with different amounts of valid semantic supervision signals in a weak-supervised or fully-supervised manner. Note that, the semantic segmentation of on-surface points and the off-surface points are both trained together with the reconstruction branch simultaneously. As shown in Figure 3.13, we provide the quantitative results of both branches. Except for a pure reconstruction model (0% in Figure 3.13). We additionally trained 6 models with randomly sampled semantic supervision signals with a ratio ranging of [100%, 10%, 1%, 0.1%, 0.01%, 0.001%]. To better demonstrate the results, we provide qualitative results in Figure 3.12 with only 0.1% supervision signals visible during training. From all these above, we can see that:

- First, the quality of surface reconstruction is very stable and consistent across the different amounts of supervision signals even if the network is trained

jointly without any pre-training. For the three datasets, the Chamfer-L1 Distance and F-score (δ) only fluctuate within a tiny range of 0.024 and 0.029 separately, which proves that our superior reconstruction ability will barely be affected by the joint optimization.

- Second, it is also interesting the quality of semantic segmentation is very stable even though the visible supervision signals decrease from 100% to only 1%, where the mIoU only decreases within 3% across all the datasets. This clearly shows that the implicit surface-oriented segmentation module is essentially aware of the sparse local features and scarce semantic annotations, which is a very important property for applications in the real world on imperfect scans.

2) Does surface reconstruction benefit the semantic branch?

From the first group of experiments, we can see that the quality of the semantic segmentation is consistently superior under challenging cases such as only very scarce labels available. In that, we conduct the second group of experiments to further explore how the surface reconstruction branch and semantic segmentation branch benefit each other. In particular, the most natural way will be to remove the reconstruction loss with unsigned distance fields regression and solely supervise the surface-oriented segmentation branch and subsequently compare this model with other models that are trained jointly with the reconstruction branch.

Metric	mIoU					
	ScanNet		2D-3D-S		SceneNN	
Recon.	w/o	w/	w/o	w/	w/o	w/
10%	**0.404**	0.401	0.602	**0.604**	0.393	**0.396**
1%	0.384	**0.392**	0.567	**0.568**	0.365	**0.371**
0.1%	0.351	**0.366**	0.473	**0.477**	0.328	**0.337**
0.01%	0.261	**0.281**	0.304	**0.325**	0.245	**0.279**
0.001%	0.205	**0.231**	0.241	**0.261**	**0.184**	0.182

Table 3.4: **Quantitative results of joint optimization** with different amounts of supervision signals and training settings. w/o in the table means that the network is trained without the reconstruction branch and vice versa.

As shown in Table 3.4, the quality of semantic segmentation is consistently better when the reconstruction branch is combined and jointly optimized, especially when

the visible semantic supervision signals decrease to ($\leq 1\%$). We hypothesise that this is because of the strong geometric priors provided by our range-aware unsigned distance fields function learning, especially the continuity in spatial regions, which will strengthen awareness of the implicit surface by coupling with implicit surface-oriented segmentation branch and help the network to deduce and propagate the 'implicit' underlying semantics of continuous surfaces to a wider area based on sparse visible semantic information.

Detailed results of semantic segmentation: We also provide more detailed results on each class on the three real-world datasets of ScanNet [5], SceneNN [11] and 2D-3D-S [12] as in Table A.1 to Table A.1 of Appendix. We also demonstrate extra results in Appendix and the provided demo video A.1.

3.4.6 Ablation Study

For the ablation study, we adopt the three real-world datasets for 3D semantic surface reconstruction. Note that, our framework is simple yet efficient and is not restricted to any specific backbone. The spherical query KPConv [94] or voxel-based backbone such as Conv3D are also applicable. However, it is shown in the experiments section that such voxelization will be extremely slow due to the reliance on the sliding window strategy and inefficient compared to our method. As a result, we keep the point-wise backbone in the ablation study. Moreover, the key components of our method are the range-aware neural interpolation module and the implicit surface-oriented segmentation module. Specifically, to argue the design insights and principles we claimed in the Section 3.2.2 and Section 3.2.3, we conduct four groups of ablation study as below:

- First, we remove $(q - p_k)$ in Equation 3.2 to analyze the importance of explicit range-aware guidance.
- Second, we add the absolute position of the query point q to Equation 3.4 for the surface-oriented semantic segmentation module.
- Third, we test the effect of different neighbour number for the KNN algorithm.

- Fourth, we further discuss the importance of the uncertainty loss used for weighting two branches automatically.

- Fifth, we also discuss the impact of colour and surface point density.

Settings	$\mathrm{CD}-L_1$	$\mathrm{FS}-\delta$	mIoU
w/o $(q-p_k)$ in Eq. 3.2	0.324	0.856	0.407
w/ q in Eq. 3.4	<u>0.300</u>	0.872	0.392
K = 1	0.313	0.850	0.396
K = 8	<u>0.300</u>	<u>0.872</u>	0.400
K = 16	0.305	0.866	0.409
w/o uncertainty loss	0.301	0.868	0.399
RangeUDF (Full)	**0.298**	**0.876**	**0.411**

Table 3.5: **Quantitative ablation study** of 3D semantic surface reconstruction on ScanNet [5] dataset and comparison to our full model.

As shown in Table 3.5, we can find that: 1) The reconstruction quality (CD-L1 and FS-δ) drops immediately by removing the explicit range-aware term of $(q-p_k)$ in Equation 3.2, which is simple yet crucial important for our methods. 2) By adding the absolute position of the query point q into Equation 3.4, the segmentation quality sharply drops to the worst one with only 0.392 in mIoU, which indicates the significance of our implicit surface-oriented segmentation module, which depends on the idea of surface patch oriented classification. 3) By changing the number of nearest neighbours, our performance is consistently superior, which demonstrates the robustness of our method. 4) In Table 3.6 and Table 3.7, we provide detailed results of our fifth experiment on the three real-world datasets. Naturally, enlarging the input points number to 50K and having more information as input such as colour will further help us to produce more promising and solid results.

3.5 Discussion and Conclusion

Conclusion: In this chapter, we propose RangeUDF, a new implicit neural representation based framework for 3D semantic surface reconstruction and scene understanding from sparse point clouds, which is no longer restricted to any type of

Tasks	Semantic Segmentation								Reconstruction							
Color	w/o RGB				w/ RGB				w/o RGB				w/ RGB			
Metrics	OA (%)		mIoU (%)		OA (%)		mIoU (%)		Chamfer-L_1		FS-0.005		Chamfer-L_1		FS-0.005	
Points	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K
0.001%	67.8	71.5	23.1	28.4	70.9	73.4	23.4	29.1	0.309	0.255	0.860	0.925	0.301	0.258	0.865	0.919
0.01%	68.7	79.7	28.1	39.0	74.1	80.1	30.9	41.7	**0.297**	**0.253**	**0.875**	0.929	0.295	0.262	0.876	0.916
0.1%	76.8	82.2	36.6	47.6	79.6	82.8	39.5	48.7	0.306	0.258	0.872	**0.930**	0.290	0.251	0.881	0.931
1%	79.5	83.1	39.2	50.0	81.5	82.5	41.9	49.5	0.302	0.260	0.870	0.917	**0.284**	0.268	**0.894**	0.917
10%	79.4	83.2	40.1	50.8	**81.8**	83.8	42.7	50.7	0.303	0.266	0.869	0.922	0.296	**0.248**	0.875	**0.935**
100%	**79.6**	**83.5**	**41.1**	**50.1**	81.6	**84.3**	**44.0**	**51.1**	0.298	0.264	0.872	0.912	0.294	0.261	0.876	0.917

Table 3.6: **Quantitative comparisons of 3D semantic surface reconstruction** from sparse point clouds on the ScanNet [5] dataset. The best results and the second-best ones are bold and underlined separately.

Datasets	2S-3D-S								SceneNN							
Tasks	Semantic Segmentation				Reconstruction				Semantic Segmentation				Reconstruction			
Metrics	OA (%)		mIoU (%)		Chamfer-L_1		FS-0.005		OA (%)		mIoU (%)		Chamfer-L_1		FS-0.005	
Points	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K	10K	50K
0.001%	61.5	65.2	26.1	30.6	0.334	0.315	0.835	0.857	72.5	75.9	18.2	21.5	0.355	0.304	0.815	0.853
0.01%	68.2	76.6	32.5	46.4	0.340	**0.312**	0.830	**0.869**	80.5	84.3	27.9	31.7	0.332	0.310	**0.833**	0.851
0.1%	75.8	83.2	47.7	61.8	0.335	0.314	0.836	0.867	84.5	86.3	33.7	40.4	**0.331**	0.299	0.830	0.866
1%	82.0	86.3	57.8	66.5	**0.332**	0.315	**0.840**	0.864	86.2	**89.1**	37.1	43.2	0.341	0.303	0.810	0.863
10%	83.6	86.3	60.4	**67.7**	0.338	0.315	0.832	0.860	86.9	87.9	**39.6**	43.4	0.336	**0.294**	0.824	**0.884**
100%	**84.1**	**86.7**	**60.8**	66.5	0.333	0.314	0.836	0.866	**87.0**	87.9	39.2	**43.8**	0.333	0.303	0.831	0.865

Table 3.7: **Quantitative comparisons of 3D semantic surface reconstruction** from sparse point clouds on the SceneNN [11] and 2D-3D-S [12] dataset. The best results and the second-best ones are in bold and underlined separately.

topologies. Our key components are a range-aware unsigned distance fields function with the neural interpolation module to solve the surface ambiguity problem that all the existing methods [6, 30] suffer from, which causes over-smooth reconstruction and bad generalization ability, as introduced in Section 2.2 of Chapter 2, and an implicit-surface oriented semantic segmentation module for learning semantics for implicit representations which no prior work can achieve. Benefiting from these designs, our method achieves state-of-the-art and shows absolute superiority in 3D surface reconstruction quality and generalization capability even in cross-domain settings, which is unprecedented for all the existing approaches. Furthermore, our method is the only one that can directly reconstruct 3D semantic surfaces from real-world sparse point clouds without any preprocessing, while existing approaches [6, 30] focus on object-level or synthetic scene reconstruction with an inefficient sliding window strategy.

Limitation and Future Work: Instead of relying on the off-the-shelf meshing algorithms such as the Ball-Pivoting algorithm and Marching Cubes, it is desirable for us to design a unique meshing strategy for implicit surface extraction from the pre-

dicted unsigned distance fields and implicit representations. Moreover, integrating instance information will also be interesting and important for applications in AR, VR and home robotics especially when it comes to scene decomposing and editing in the future, which is not yet explored by any researchers for now.

Discussion: It is worth noting that our RangeUDF is very straightforward in addressing the open problems of existing methods such as surface ambiguity. It is simple yet presents extraordinary and unprecedented state-of-the-art results, pushing forward the reconstruction accuracy to the next level. Therefore, we hope that the novelty and usefulness of this work can be an insight for other researchers.

CHAPTER 4

Explicit Representation Learning for Parametric Reconstruction

As a typical branch of explicit representation learning, mesh representation is commonly used to represent a single topology such as humans and animals for pose estimation and shape recovery and clothing [104–106]. Despite the limitations of topologies and details of the parametric models such as SMPL model [1] and SMPL-X model [2], parametric 3D reconstruction is becoming more important these years due to its wonderful generalization ability and geometry priors. In terms of parametric 3D human reconstruction, which is a trending direction for augmented reality (AR), and virtual reality (VR) applications, recovering two hands from monocular RGB images is the most challenging part due to frequent occlusion and mutual confusion. Existing methods mainly learn an entangled representation to encode two interacting hands, which are incredibly fragile to impaired interaction, such as truncated hands, separate hands, or external occlusion.

This chapter presents Attention Collaboration-based Regressor (ACR) [26], which makes the first attempt to reconstruct hands in arbitrary scenarios. To achieve this, ACR explicitly mitigates interdependencies between hands and between parts by leveraging center-based and part-based attention for feature extraction. However, reducing interdependence helps to release the input constraint while weakening the

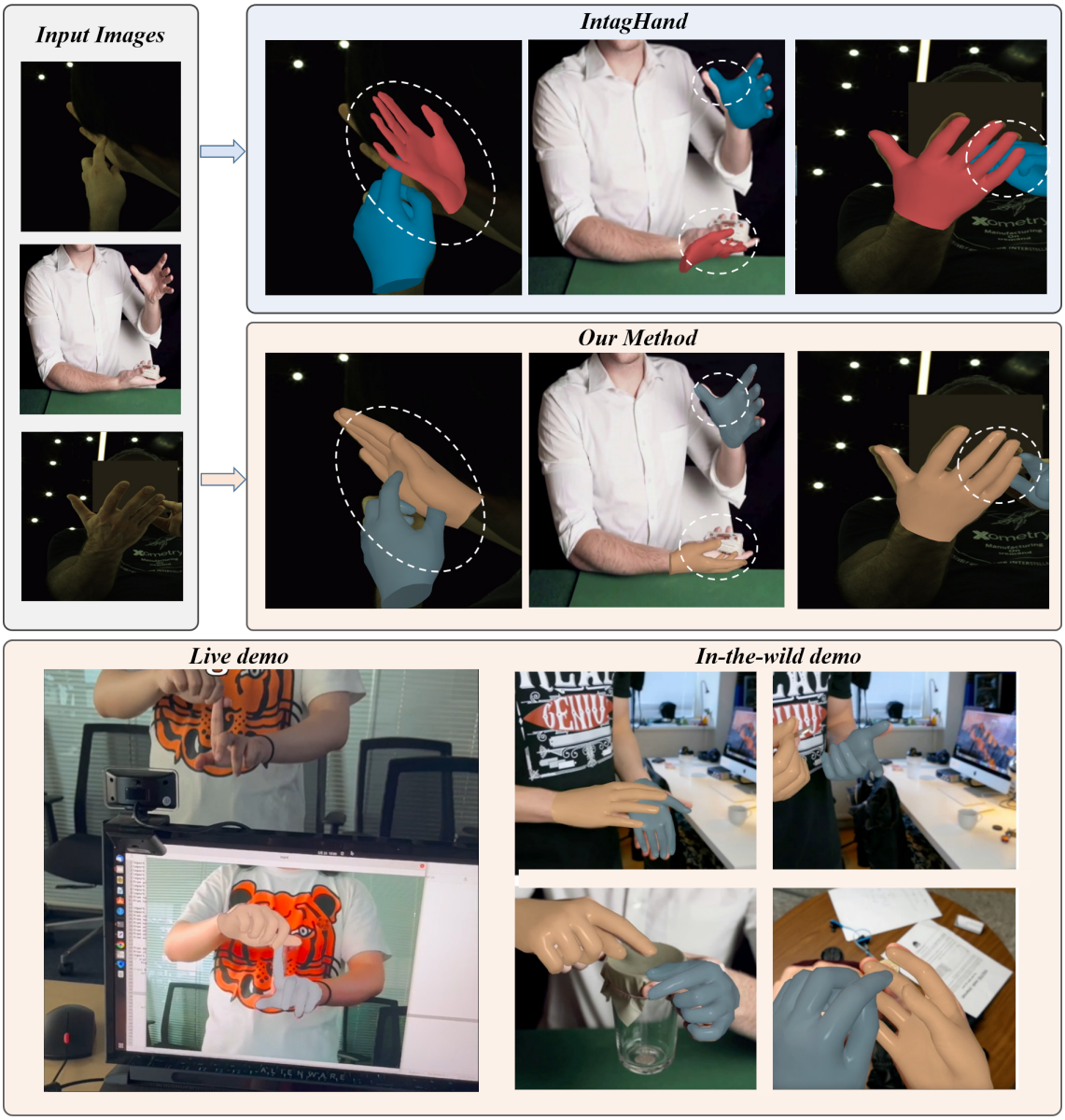


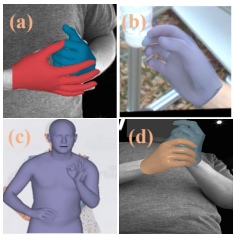
Figure 4.1: **Motivating example**. ACR makes the first attempt to reconstruct hands under arbitrary scenarios by representation disentanglement and interaction mutual reasoning while the previous state-of-the-art method IntagHand [8] failed.

mutual reasoning about reconstructing the interacting hands. To model interacting hands better, ACR also learns a cross-hand prior for handling the interacting hands better based on center attention. We evaluate our method on various types of hand reconstruction datasets. Our method significantly outperforms the best interacting-hand approaches on the InterHand2.6M dataset while yielding comparable performance with the state-of-the-art single-hand methods on the FreiHand dataset. More qualitative results on in-the-wild and hand-object interaction datasets and web images/videos further demonstrate the effectiveness of our approach for arbitrary hand reconstruction.

4.1 Introduction

3D hand pose and shape reconstruction based on a single RGB camera plays an essential role in various emerging applications, such as augmented and virtual reality (AR and VR), human-computer interaction (HCI), 3D character animation for movies and video games, etc. However, this task is very challenging due to limited labelled data, occlusion, depth ambiguity, etc. Earlier attempts [57, 107–109] level down the problem difficulty and focus on single-hand reconstruction. These methods started from exploring weakly-supervised learning paradigms [57] to design more advanced network models [110]. Although single-hand approaches can be extended to reconstruct two hands, they generally ignore the inter-occlusion and confusion issues, thus failing to handle two interacting hands.

To this end, recent research focus has shifted towards reconstructing two interacting hands. Wang et al. [9] extract multi-source complementary information to reconstruct two interacting hands simultaneously. Rong et al. [70] and Zhang et al. [14] first obtain initial prediction and stack intermediate results together to refine two-hand reconstruction. The latest work in the field by Li et al. [8] gathers pyramid features and two-hand features as the input for a GCN-based network that regresses two interacting hands as a whole. These methods share the same principle: treating two hands as an integral and learning a unified feature to refine or regress the interacting-hand model ultimately. The strategy delivers the advantage



Method Type	(a) Interacting	(b) Single	(c) Full-body	(d) Ours
Interacting Hands	✔	✗	✗	✔
Detector Free	✗	✗	✗	✔
Truncated Hands	✗	✗	✗	✔
Object occlusion	✗	✗	✗	✔

Figure 4.2: Comparison: Our method has more properties that are desirable for real-world applications.

of explicitly capturing the correlation between hands but inevitably introduces the input constraint of two hands. This limitation also makes the methods particularly vulnerable and easily fail to handle inputs containing imperfect hand interactions, including truncated hands or occlusions.

This paper takes the first step towards reconstructing two hands in arbitrary scenarios. Our first key insight is leveraging center and part attention to mitigate interdependencies between hands and between parts to release the input constraint and eliminate the prediction sensitivity to a small occluded or truncated part. To this end, we propose Attention Collaboration-based Regressor (ACR). Specifically, it comprises two essential ingredients: Attention Encoder (AE) and Attention Collaboration-based Feature Aggregator (ACFA). The former learns the hand-center and per-part attention maps with a cross-hand prior map, allowing the network to be aware of the visibility of both hands and each part before the hand regression. The latter exploits the hand-center and per-part attention to extract global and local features as a collaborative representation for regressing each hand independently. In contrast to the existing method, our method provides more advantages, such as being hand detector-free and the ability to adapt to arbitrary scenarios that all the other methods cannot perform well such as occlusion, truncation, and compatibility with full-body capture. Furthermore, experiments show that ACR achieves lower error on the InterHand2.6M dataset [13] than the state-of-the-art interacting-hand methods, demonstrating its effectiveness in handling interaction challenges. Finally, results on in-the-wild images or video demos indicate that our approach is promising for real-world application with the powerful aggregated representation for arbitrary hand reconstruction.

Our key contributions are summarized as:

- We take **the first step toward reconstructing two hands at arbitrary scenarios**.
- We propose to **leverage both center-based and part-based representation to mitigate interdependencies between hands and between parts** and release the input constraint.
- In terms of modeling for interacting hands, we **propose a cross-hand prior reasoning module with an interaction field** to adjust the dependency strength.
- Our method **outperforms existing state-of-the-art approaches significantly on the InterHand2.6M benchmark**. Furthermore, ACR is the most practical method for various in-the-wild application scenes among all the prior arts of interacting and single hand reconstruction.

4.2 Preliminaries

Parametric Hand Model: We use a differentiable parametric model MANO [3] to represent the hand mesh, which contains a pose parameter $\theta \in \mathbb{R}^{16\times3}$ and a shape parameter $\beta \in \mathbb{R}^{10}$. We further utilize 6D representations [111] to present our hand pose as $\theta \in \mathbb{R}^{16\times6}$. The final hand mesh M could be reconstructed via a differentiable MANO model as:

$$M = S(T(\beta, \theta), J(\beta), W) \tag{4.1}$$

where $S(\cdot)$ is a skinning function, W is the skinning weight matrix, T is a parametric model template for the human hand and J is the hand joint position of shape β. Subsequently, 3D joints $J_{3D} \in \mathbb{R}^{21\times3}$ can be retrieved from the mesh: $\hat{J_{3D}} = RM$, where R is a pre-trained linear regressor and $M \in \mathbb{R}^{778\times3}$.

Weak Perspective Camera Model: To render our 3D hand mesh as an overlay on the image and 2D joints projection for weak supervision, we adopt a weak-

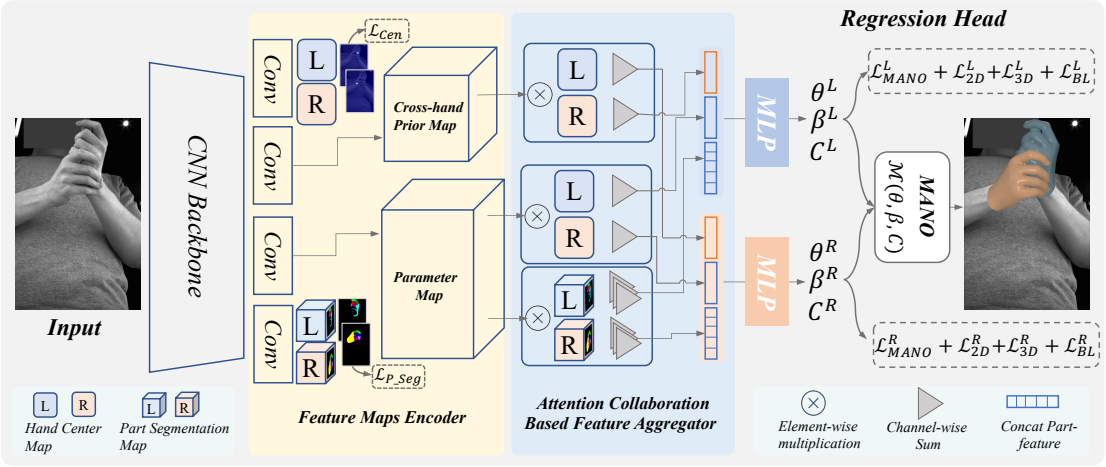


Figure 4.3: **ACR network architecture:** ACR takes a full-person image and uses a feature map encoder to extract hand-center maps, part-segmentation maps, cross-hand prior maps, and parameter maps. Subsequently, the feature aggregator generates the final feature for hand model regression based on these feature maps.

perspective camera model (s, t_x, t_y) following prior arts in human and hand parametric reconstruction [36, 57, 112]. For a set of 3D joints retrieved from the mesh $\hat{J_{3D}}$, the projected 2D joints $J_{pj2d} = (x_{pj2d}, y_{pj2d}) \in \mathbb{R}^{21\times 2}$ are represented as:

$$x_{pj2d} = sx_{d3} + t_x, \quad y_{pj2d} = sy_{d3} + t_y \tag{4.2}$$

where s is the scale and t is the translation for the 2D projection on the image plane.

4.3 Method Overview

Unlike existing works [8, 57, 113–115] that rely on an external detector to perform entangled bounding-box-level representation learning. Figure 4.3 presents the overview of our method ACR. Given a single RGB image $\mathbf{I}$ as input, ACR outputs four maps, which are the Cross-hand Prior map, Parameter map, Hand Center map, and Part Segmentation map. Based on the parameter map, which predicts weak-perspective camera parameters and MANO parameters for both left hand and right hand at each pixel, ACR then leverages three types of pixel-level representations for attention aggregation from the Parameter map. First, ACR explicitly mitigates inter-dependencies between hands and between parts by leveraging center and part-

based representation for feature extraction using part-based attention. Moreover, ACR also learns a cross-hand prior for handling the interacting hands better with our third Cross-hand Prior map. Finally, after aggregating the representations, we feed estimated parameters $\boldsymbol{F_{out}}$ to MANO [3] model to generate the hand meshes.

4.3.1 Representations of Attention Encoder

In this section, we present the details of each output map or Attention Encoder (AE) module and their representations as shown in Figure 4.3. Given a monocular RGB image $I \in \mathbb{R}^{3\times H\times W}$ as input, we first extract a dense feature map $F \in \mathbb{R}^{C\times H\times W}$ through our CNN backbone. ACR then leverages three types of pixel-level representations for robust arbitrary hand representations disentanglement and mutual reasoning under complex interaction scenarios. For simplicity, we denote the handedness by $\boldsymbol{h} \in \{L, R\}$.

Parameter map: $M_p \in \mathbb{R}^{218\times H\times W}$ can be divided into two maps for left hand and right hand separately, where the first 109 dimensions are used for left-hand feature aggregation and the rest for the right hand. For each of the map $M_p^h \in \mathbb{R}^{109\times H\times W}$. The 109 dimensions consist of two parts, MANO parameter $\theta \in \mathbb{R}^{16\times 6}$, $\beta \in \mathbb{R}^{10}$ and a set of weak-perspective camera parameters (s, t_x, t_y) that represents the scale and translation for the 2D projection of the individual hand on the image. This map serves as our **basic** module for aggregated representation learning.

Hand Center map: $A_c \in \mathbb{R}^{2\times H\times W}$ also consists of two parts for left hand and right hand, which can be represented as $A_c^h \in \mathbb{R}^{1\times H\times W}$. Each of the maps is rendered as a 2D Gaussian heatmap, where each pixel represents the possibility of a hand center being located at this 2D position and the center is defined as the center of all the visible **MCP** joints. For adaptive global representation learning, we generate heatmaps by adjusting the Gaussian kernel size K according to the bounding box size of the hand in data preparation for supervision. As the **first** representation of ACR, this map explicitly mitigates inter-dependencies between hands and serves as an attention mask for better global representation learning.

Part Segmentation map: $A_p \in \mathbb{R}^{33\times H\times W}$ is learnt as a probabilistic segmentation volume. Each pixel on the volume is a channel of probability logits over 33

classes, which consists of 1 background class and 16 hand part classes for each hand corresponding to the MANO model, so we have $A_p^h \in \mathbb{R}^{16\times H\times W}$. We obtain the part segmentation mask obtained by rendering the ground truth MANO hand mesh using a differentiable neural renderer [35]. As the **second** representation of ACR, this map serves as an attention mask for part representation learning.

Cross-hand Prior map: $M_c \in \mathbb{R}^{218\times H\times W}$ contains two maps and each of them can be written as $M_c^h \in \mathbb{R}^{109\times H\times W}$. It is split into two sets of parameters which are MANO parameter $\theta \in \mathbb{R}^{16\times 6}$, $\beta \in \mathbb{R}^{10}$ and 3 camera parameters for cross hand **inverse** feature query. Empirically, the two hands' pose will be highly correlated when they are closely interacting within the interaction field (**IF**), which is introduced in 4.3.3. As our **third** representation, aggregating this module into our robustly disentangled representations provides us with powerful mutual reasoning ability under severe interaction scenarios.

4.3.2 Robust Representation Disentanglement

Interestingly, we found all the existing prior works such as [8, 13, 14] share the same principle by treating two hands as an integral and implicitly learning an **entangled** representation to refine or regress the interacting-hand model ultimately, which will cause ambiguity and unnecessary constraints. They require that the input image must be fixed to **two closely** interacting hands and the hands must occupy the most region after cropping. As shown in Figure 4.1, we can see that (1) their methods will completely fail when the two hands are not close enough but still interacting. Second, (2) they are inherently agnostic to the individual hand with such **entangled** representation in the image thus resulting in a degeneration in both hands when one of the two hands is truncated, occluded, or duplicated.

Unlike all the existing approaches for interacting hands reconstruction [8, 13, 14], our first step towards building **arbitrary** hands representation is - **disentanglement** by decomposing the ambiguous hand representations. Thanks to the powerful pixel-wise representation of the Hand Center map, we are able to disentangle **inter**-hand dependency and build an explicitly separate feature representation for the two hands. However, when the two centers are getting too close, these feature rep-

resentations could also be highly ambiguous. Subsequently, for better-disentangled feature representation learning, inspired by [112], we adopt a collision-aware center-based representation to further split the features of two hands by applying Equation 4.3. When the two hands are too close to each other with a Euclidean distance d smaller than $k_L + k_R + 1$. The new centers will be generated as:

$$\begin{aligned} \hat{\boldsymbol{C}}_{\boldsymbol{L}} = C_L + \alpha R, \quad \hat{\boldsymbol{C}}_{\boldsymbol{R}} = C_R - \alpha R, \\ \boldsymbol{R} = \frac{k_L + k_R + 1 - d}{d}(C_L - C_R) \end{aligned} \tag{4.3}$$

where C_L, k_L and C_R, k_R stand for two hand centers and their kernel size. R means the repulsion vector from C_L to C_R. In addition, α refers to an intensity coefficient to adjust the strength. Finally, the global representation $\boldsymbol{F}_{\boldsymbol{g}}^{\boldsymbol{h}} \in \mathbb{R}^{J*C+(10+3)}$, where $C = 6$, is extracted by combing Hand Center map A_c with parameter map M_p as:

$$F_g^h = f_g(\sigma(A_c^h) \otimes M_c^h) \tag{4.4}$$

where $\sigma, \otimes$ and f_g are spatial softmax, pixel-wise multiply and a point-wise Multi-Layer Perceptron (MLP) layer separately, and $\boldsymbol{h} \in \{L, R\}$.

With such global feature representation F_g, we have successfully disentangled the inter-dependency. However, having only such global representation will lead to instability under occlusion and losing the ability to recover details, due to the unnecessary **inner** dependency of each hand part. Subsequently, we need to further disentangle our representation utilizing our Part Segmentation map A_p following [36]. For simplicity, we ignore the $\boldsymbol{h} \in \{L, R\}$ here, the two hands follow the same formulation as:

$$F_p^{(j,c)} = \sum_{h,w} \sigma(A_p^j) \odot M_p^c, \tag{4.5}$$

where $F_p \in \mathbb{R}^{J \times C}$ is the final part representation and $F_p^{(j,c)}$ is its pixel at (j, c). $\odot$ is the Hadamard product. Subsequently, the part segmentation maps after spatial softmax normalization σ are used as soft attention masks to aggregate features in M_p^c. We follow prior arts to implement a dot product-based method by reshaping

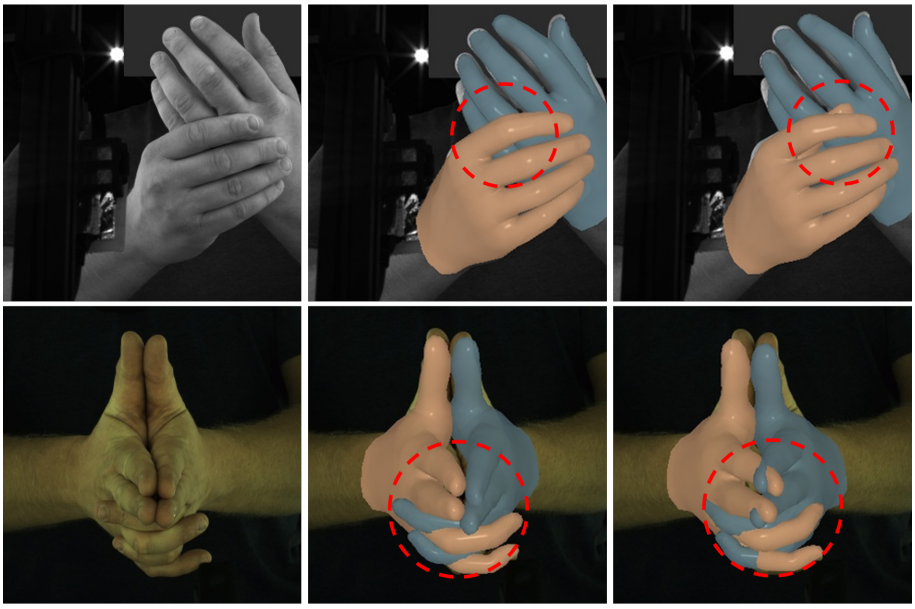


Figure 4.4: **Effect of Mutual Reasoning**. It is shown that our mutual reasoning module explicitly helps to deduce and recover the correlation between closely interacting hands with less mutual occlusion.

the tensor at first: $F_p = \sigma(A_p^*)^T M_p^*$, where $M_p^* \in R^{HW \times C}$ and $A_p^* \in R^{HW \times J}$ are the parameter map M_p and reshaped part segmentation A_p without background mask. Finally, the global feature representation F_g and part representation and F_p are aggregated into our robust inter and inner disentangled representation.

4.3.3 Mutual Reasoning of Interaction

Despite the powerful disentangled representations, it has been explored that the states of two interacting hands are highly correlated [8, 14] when they are interacting closely. Simply disentangling inter and inner dependencies as the final representation will weaken the mutual reasoning about reconstructing the interacting hands. Subsequently, we design a novel mutual reasoning strategy by reusing the center-based attention via a **inverse query**:

$$\begin{aligned} F_c^{R \rightarrow L} &= f_c(\sigma(A_c^R) \otimes M_c^L), \\ F_c^{L \rightarrow R} &= f_c(\sigma(A_c^L) \otimes M_c^R), \end{aligned} \tag{4.6}$$

where $F_c^{R\rightarrow L}$ is the left-hand prior representation that is deduced from right-hand attention and vice versa. M_c is the output dense feature map from cross-hand-prior attention blocks, A_c is our center based attention map, and L, R stand for left hand and right hand. $\sigma, \otimes$ and f_c are spatial softmax, pixel-wise multiply and a point-wise MLP layer.

However, for two more distant hands or a single hand, the correlation between them should be mitigated or eliminated. Subsequently, we also propose a new mechanism, interaction field (**IF**) to adjust the dependency strength. Specifically, by first computing the Euclidean distance $\boldsymbol{d}$ between the hands, when the two hands are too close to each other and entering the field of **IF**$= \gamma(k_L + k_R + 1)$, where γ is a field sensitivity scale. The interaction intensity coefficient $\boldsymbol{\lambda}$ will be computed as:

$$\boldsymbol{\lambda}_{(}\boldsymbol{C_L}, \boldsymbol{C_R}) = \begin{cases} 0, & d > IF \\ \frac{IF-d}{d}||C_L - C_R||_1, & d <= IF \end{cases}$$

The interaction intensity coefficient $\boldsymbol{\lambda}$ helps our cross-hand prior representation to formulate an adaptive interaction field that can better model the correlations of two hands while keeping sensitive to close interaction and separation to avoid unnecessary feature entanglement. Finally, our final output self-adaptive robust representation could be represented as:

$$F_{out}^h = f_{out}(concat(F_g^h, F_p^{h*}, \lambda F_c^h)) \tag{4.7}$$

where f_{out} is point-wise MLP layers for regressing the final representation $F_{out}^h \in \mathbb{R}^{109}$, and $F_c^{h*} \in \mathbb{R}^{J*C}$ is reshaped part disentangled representation. Finally, the regressed parameters F_{out}^h are fed into the MANO model to generate the final mesh.

4.3.4 Loss Functions

For training ACR with three types of powerful representation, our loss functions are divided into three groups, as demonstrated in Fig 4.3. Specifically, ACR is supervised by the weighted sum of all loss items for both the left hand and the right

hand: mesh recovery loss, center-based attention loss, and part-based attention loss.

Center Attention Loss can be treated as a segmentation problem, however, the Gaussian distribution on the image is a relatively small area and there is an imbalance between the positive and negative samples. Subsequently, we utilize focal loss [116] to supervise our center map regressor as:

$$\boldsymbol{\mathcal{L}_c} = \sum_{h \in \{L,R\}} f(A_c^h, \hat{A}_c^h), \tag{4.8}$$

where f is focal loss [116], $h \in \{L, R\}$ means left hand and right hand, and $\hat{A}_c^i$ is the ground truth hand center map for hand type i. For simplicity, here we abbreviate the formulation of focal loss.

Part Attention Loss is used to supervise our Part-based Representation learning. We only supervise this loss with CrossEntropy loss in the first 2 epochs and continue to train with other losses until it converges.

$$\boldsymbol{\mathcal{L}_{seg}} = \frac{1}{HW} \sum_{h,w} CrossEntropy(\sigma(A_p^{hw}), A_p^{\hat{hw}}), \tag{4.9}$$

where $\hat{A}_p$ means GT part segmentation maps and $A_p^{\hat{hw}}$ is the ground truth class label at the location of (h, w). Different from our part soft attention mask, $A_p^{hw} \in \mathbb{R}^{33\times1\times1}$ here means the probabilistic segmentation volume at the pixel position of (h, w) and σ means softmax along channel dimension. Note that we do not need to omit the background class here.

Mesh Recovery Loss is applied for each hand, thus we ignore the handedness $\boldsymbol{h} \in \{L, R\}$ here for simplicity. Finally, the loss for the left hand and the right hand will be summed into the total loss. Instead of relying on the ground truth vertex positions, which could cause degeneration in generalization ability, we decouple our mesh loss into 3 parts:

$$\boldsymbol{\mathcal{L}_{mesh}} = \boldsymbol{\mathcal{L}_{mano}} + \boldsymbol{\mathcal{L}_{joint}}, \tag{4.10}$$

where $\mathcal{L}_{mano}$ is the weighted sum of $L2$ loss of the MANO parameters θ and β, namely $w_\theta \mathcal{L}_\theta + w_\beta \mathcal{L}_\beta$:

$$\boldsymbol{\mathcal{L}_{\theta}} = w_{\theta}||\theta - \hat{\theta}||_2^2, \quad \boldsymbol{\mathcal{L}_{\beta}} = w_{\beta}||\beta - \hat{\beta}||_2^2, \tag{4.11}$$

$\mathcal{L}_{joint}$ is the weighted sum of $\mathcal{L}_{3D}$, $\mathcal{L}_{2D}$ and a bone length loss $\mathcal{L}_{bone}$ to provide better geometric constraint to the reconstructed mesh, which is computed by $L2$ distance between i^{th} ground truth bone length $\hat{b_i}$ and predicted length b_i:

$$\begin{aligned}
\boldsymbol{\mathcal{L}_{3D}} &= w_{j3d}\mathcal{L}_{MPJPE} + w_{paj3d}\mathcal{L}_{PA-MPJPE}, \\
\boldsymbol{\mathcal{L}_{PJ2D}} &= w_{pj2d}||PJ_{2D} - \hat{J_{2D}}||_2^2, \\
\boldsymbol{\mathcal{L}_{bone}} &= \sum_i ||b_i - \hat{b_i}||_2^2,
\end{aligned} \tag{4.12}$$

where $\mathcal{L}_{MPJPE}$ is the $L2$ loss between ground-truth 3D joints $\hat{J_{3D}}$ and predicted ones J_{3D} retrieved from predicted mesh. $\mathcal{L}_{PA-MPJPE}$ is computed as the Procrustes-aligned mean per joint position error (PA-MPJPE). We do not supervise camera parameters directly, instead, the network adjusts the camera parameters by computing the $L2$ loss between ground truth $\hat{J_{2D}}$ and the projected 2d joints PJ_{2D} retrieved by a weak-perspective camera: PJ_{2D} as $x_{pj2d} = sx_{3D} + t_x, y_{pj2d} = sy_{3d} + t_y$. Finally, to compute $\mathcal{L}_{mesh}$ as a weighted sum, we apply $w_{j3d} = 200$, $w_{paj3d} = 360$, $w_{pj2d} = 400$, $w_{bl} = 200$. For $\mathcal{L}_{mano}$, we use $w_{pose} = 80$, $w_{shape} = 10$ in our experiments.

Total Loss is the weighted sum of the described loss above and can be represented as:

$$\boldsymbol{\mathcal{L}_{total}} = \boldsymbol{\mathcal{L}_{mesh}} + w_c\boldsymbol{\mathcal{L}_{c}} + w_p\boldsymbol{\mathcal{L}_{seg}}, \tag{4.13}$$

where $w_c = 160$, $w_p = 160$ and $\mathcal{L}_{mesh}$ is already a weighted sum. Each part is activated only when the corresponding ground truth is available. Utilizing 2D keypoints can further fill the gap between real-world and in-the-lab data and enable us to train on 2D datasets in a weak supervision manner. When the mano annotation is valid, all parts of the loss will be activated. Finally, all of these losses are trained simultaneously in an end-to-end manner.

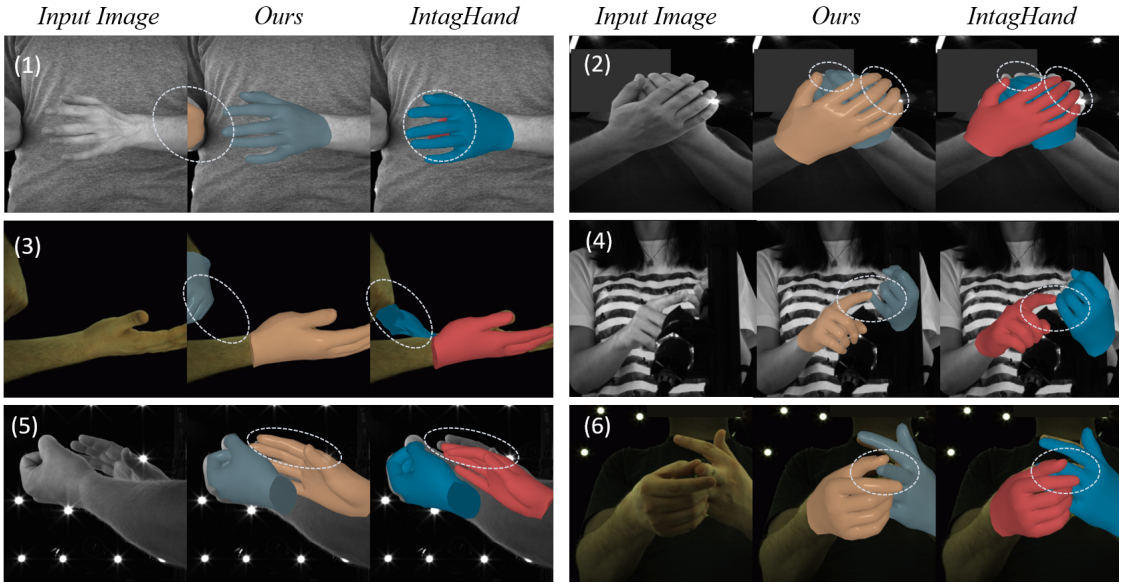


Figure 4.5: Qualitative comparison with on InterHand 2.6M test dataset. Our approach generates better results in two-hand reconstruction, particularly in challenging cases such as external occlusion (1), truncation (3-4), or bending one finger with another hand (6). More results can be found in the appendix

4.4 Implementation

Backbone network: We implement our network based on PyTorch [98]. For the backbone network, we have trained with both ResNet-50 [117] and HRNet-W32 [118], for faster inference speed or better reconstruction results respectively. Unlike existing approaches that require an external hand detector, our method can reconstruct arbitrary hands in an end-to-end manner without any extra information needed. Furthermore, our method does not limit its input to two-hand. Given a monocular raw RGB image without cropping or detection, all the input raw images and segmentation maps are resized to 512×512 while keeping the same aspect ratio with zero padding, then we extract the feature maps $f \in R^{(C+2)\times H\times W}$ from the backbone network with CoordConv [119]. The feature maps are fed finally to four Conv blocks to produce the four maps for representation aggregation.

Training: For comparison on the InterHand2.6M dataset, we train our model using Adam optimizer with a learning rate 5e-5 for eight epochs. We do not supervise L_{seg} and L_{MANO} when there is no MANO label valid. Because our ground truth segmentation is obtained from rendering ground truth MANO hand mesh using a

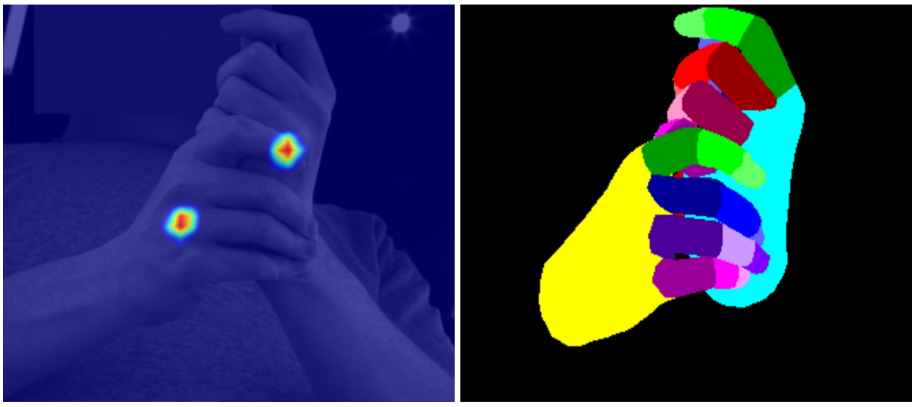


Figure 4.6: Illustration of hand center and hand part segmentation.

neural renderer [35]. For all of our experiments, we initialized our network using the pre-trained backbone of HRNet-32W from [113] to speed up the training process. We train our network using 2 V100 GPUs with *batchsize* of 64. The size of our backbone feature is 128×128 and the size of our 4 pixel-aligned output maps is 64×64. We applied random scale, rotation, flip, and colour jitter augmentation during training.

Testing: For all the experiments, if not specified, the backbone is HRNet-32W. For comparison with state-of-the-art, we use the full official test set for evaluation. The confidence threshold is set to 0.25 with a max detection number of one left hand and one right hand, as we only have one left hand and one right hand in all the training and testing sets.

Global representation: To guide our network to gain a better global representation, we adopt a scale-adaptive Gaussian kernel for our center map generation. As shown in Figure 4.6. To generate a scale adaptive Gaussian heatmap, we adopt a Gaussian kernel according to the size of the bounding box. The bounding box is roughly computed by the maximum and minimum values of the visible keypoints of the hand. Specifically, the center-based attention is represented by a Gaussian map where its kernel size K is computed according to the hand box. Let d be the diagonal length of the box, W_b be the width, then the kernel size to generate the supervision map can be computed by

$$k = k_{min} + \delta_k \times (\frac{d}{\sqrt{2}W})^2, \tag{4.14}$$

where k_{min} stands for the minimum kernel size and we would adjust kernel size depending on the different hand scale. δ is the adjusting factor to control the expanding size of the kernel size. In all of our experiments, we set $k_{min} = 2$ and $\delta_k = 7$ as the default setting.

Part representation: Our ground truth segmentation map is rendered by utilizing the ground truth MANO mesh and camera parameters provided by InterHand or FreiHand with a neural renderer [35], thus we only supervise the part segmentation branch when the ground truth MANO parameter or the ground truth segmentation is available. Our segmentation map is represented on the right of Figure 4.6. The background class is 0 (black part). The labels for left-hand parts are from $1 \sim 16$ and right-hand labels are $17 \sim 32$.

4.5 Experiments

In this work, we use four metrics to evaluate the reconstruction quality of our method, which are MPJPE, MPVPE, PA-MPJPE, and PA-MPVPE. Please note that all of the evaluation metrics are performed after **root joint alignment** of each hand. It is worth noting that our concurrent work and prior arts [8, 14] typically need to recover the mesh to ground truth scale by using **extra ground truth information** during evaluation, which is not fair for previous methods. For a fair comparison with them, we compare both the correct protocol as the previous methods and their 'unfair' protocol of using extra ground truth scale and box.

4.5.1 Metrics

Following prior works [8, 14], we adopt four evaluation metrics as below:

MPJPE measures the mean per joint position error in millimetres, which is the mean Euclidean distance between the predicted 3D joint locations to ground truth 3D joint locations after root joint alignment.

MPVPE measures the mean per vertex position error in millimetres. The average Euclidean distance between the hand mesh predictions and the ground truth MANO hand mesh after aligning them by root joint.

PA-MPJPE is the MPJPE after Procrustes alignment. By Procrustes aligning the predictions and the ground truth mesh, it eliminates the effects of translation, rotation and translation and focuses on the reconstruction accuracy.

PA-MPVPE is the MPVPE after Procrustes alignment. By Procrustes aligning the predicted mesh and the ground truth mesh, it eliminates the effects of translation, rotation and translation.

4.5.2 Datasets

InterHand2.6M [13] is the first one and the only publicly available dataset for two-hand interaction with accurate two-hand mesh annotations. This large-scale real-captured dataset, with both accurate human (H) and machine (M) annotated 3D pose and mesh annotation, contains 1,361,062 frames for training and 849,160 frames for testing, and 380,125 for validation in total. These subsets are split into two parts: interacting hands (IH) and single hand (SH). We use the 5 FPS IH subset with H+M annotations for our experiments.

FreiHand [15] is a single hand 3D pose estimation dataset. For each frame, it has MANO annotation and 3D keypoints annotation. There are 4×32,560 frames for training and 3960 frames for evaluation and testing. The initial sequence with 32560 frames is captured with a green screen background, allowing background removal.

RGB2Hands [9] is an RGB dataset to evaluate the interacting two hands. It has 4 sequences: crossed, occlusion, shuffle and scratch. It has hand joint labels on paired RGB and depth images. We use it for qualitative evaluation in our work.

EgoHands Datasets[120] contains 48 egocentric video sequences recorded in the real world with two-person interactions in different scenes. This dataset contains both two-hand interactions and two-person interactions such as playing chess and puzzle, which are not constrained to only two hands. However, it has no mesh annotation, thus we only use it for qualitative evaluation.

	extra info.	IH MPJPE	IH MPVPE	SH MPJPE	SH MPVPE
(-) Zimmermann et al.[115]	Box	36.36	-	-	-
(-) Zhou et al.[109]	Box	23.48	23.89	-	-
(-) Boukhayma et al.[57]	Box	16.93	17.96	-	-
(-) Spurr et al. [121]	Box	15.40	-	-	-
Moon et al. [13]	Box	16.02	-	12.16	-
Fan et al. [113]	Box	14.27	-	11.32	-
Zhang et al. [14]	Box	13.48	13.95	-	-
IntagHand [8]	Box	10.27	10.53	9.67	9.91
Ours	**-**	**9.08**	**9.31**	**6.85**	**7.01**
IntagHand [8]	Box+scale	9.40	9.68	9.0	9.18
Ours	scale	**8.41**	**8.53**	**6.09**	**6.21**

Table 4.1: Comparison with state-of-the-art on InterHand2.6M[13]. (-) means single hand reconstruction method. Except for our approach, all the others use ground-truth bounding boxes from the dataset. The single-hand results are taken from [14]. We report results on the official test split of the InterHand2.6M dataset for a fair comparison. We noted that the reported result of IntagHand is obtained from a filtered test set. We, therefore, get the result on the standard test set by running its released code [14].

4.5.3 Comparison to State-of-the-art Methods

Results on InterHand2.6M and FeiHand datasets: We first compare our method with single-hand and interacting-hand approaches on InterHand2.6M. We report results on the official test split of the InterHand2.6M dataset for a fair comparison. As the reported result in the paper of IntagHand is obtained from a filtered test set, we get the result on the standard test set by running its official code. Tab 4.1 presents comparison results on the **I**nteracting hands (IH MPJPE), and **S**ingle hand (SH MPJPE) subset, and the full-set (MPJPE). Not surprisingly, we can observe that single-hand methods generally perform poorly on the IH subset, as their method designs dedicate to single-hand input. Next, we perform a comparison with two state-of-the-art interacting-hand approaches [14] and [8]. The first one adopts a refinement strategy that predicted the initial pose and shape from deeper features and gradually refined the regression with lower-layer features. The latter IntagHand incorporates pyramid features with GCN-based to learn implicit attention to address occlusion and interaction issues, while IntagHand is our concurrent work and outperforms [14]. However, our proposed method constantly surpasses IntagHand without any extra information needed. Specifically, our method obtained the lowest

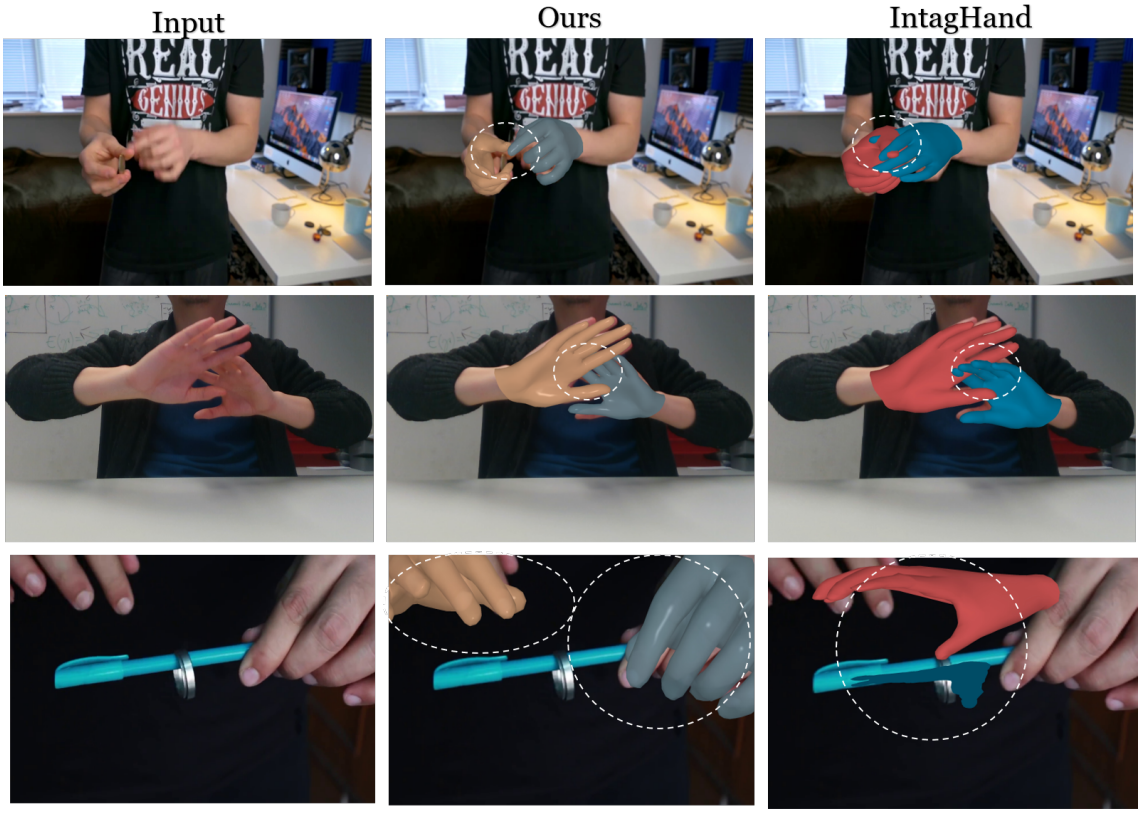


Figure 4.7: Qualitative comparison with IntagHand [8] on in-the-wild images.

Method	PA-MPJPE	PA-MPVPE
Mesh Graphormer[122]	6	5.9
METRO[123]	6.8	6.7
I2L-MeshNet[124]	7.4	7.6
HandTailor[114]	8.2	8.7
ours	6.9	7.0

Table 4.2: Comparison with state-of-the-art on FreiHand [15] Benchmark.

MPJPE of 8.41 on the IH subset, demonstrating its effectiveness in handling interacting hands. It also achieves a 6.09 MPJPE on the SH dataset that outperforms IntagHand by a large margin, showing our method remains superior on single-hand reconstruction.

We also compare our method with single-hand methods on the single-hand dataset FreiHand [15]. As shown in Table 4.2, the transformer-based method achieves the best result. Nevertheless, our method obtains comparable performance to this state-of-the-art single-hand approach, revealing its potential to improve single-hand reconstruction.

Qualitative Evaluation: We previously demonstrated our method significantly outperforms IntagHand in quantitative experiments. To gain insight into this result, this section provides more qualitative results on in-the-wild datasets or web videos (watch video ***acr_in_the_wild.mp4*** for more detail). First, we compare our method with the previous state-of-the-art, IntagHand[8] on RGB2Hands [9] and Ego2Hand datasets [120]. We also provide a qualitative comparison of two approaches on web videos (obtained from YouTube). Since IntagHand can only deal with well-cropped hand regions, we acquire the result by employing them with a hand detector to crop out the hand region from an in-the-wild image while keeping the aspect ratio. For visualization, we directly project the rendering results of IntagHand back to the **original image** instead of the cropped image because our method does not need to crop and is adaptive to any image resolution.

We conduct a qualitative comparison between IntagHand and our method. Interestingly, our approach generally produces better reconstruction results than IntagHand in almost all cases, especially challenging cases such as external occlusion, truncated hands and separate hands. Figure 4.5 shows some examples of these cases. This result indicates that our method for two-hand reconstruction is less sensitive to some impaired observation. We also try our method to reconstruct in-the-wild images containing single hand, ego-view, hand-object interaction, and truncated hands. Figure 4.7 presents some representative images where hands are accurately reconstructed, proving that our method has strong generality and is very promising for real-world applications.

Our method performs better than IntagHand under nearly all cases, particularly in challenging cases such as truncated hands, severe occlusion, and hand-object interaction. Fig 4.10 and 4.11 show some representative examples. For instance, if one hand is severely impeded by the other hand or separated (like Figure 4.10(c)), our approach yields more reasonable results than IntagHand. Another case that IntagHand usually fails to handle is truncation (i.e., 4.11(f), hand pars truncated by image boundary). In contrast, our method built on part-level representation learning is more stable in this situation. Moreover, our approach also performs much better than IntagHand on hand-object interaction data. This is because IntagHand

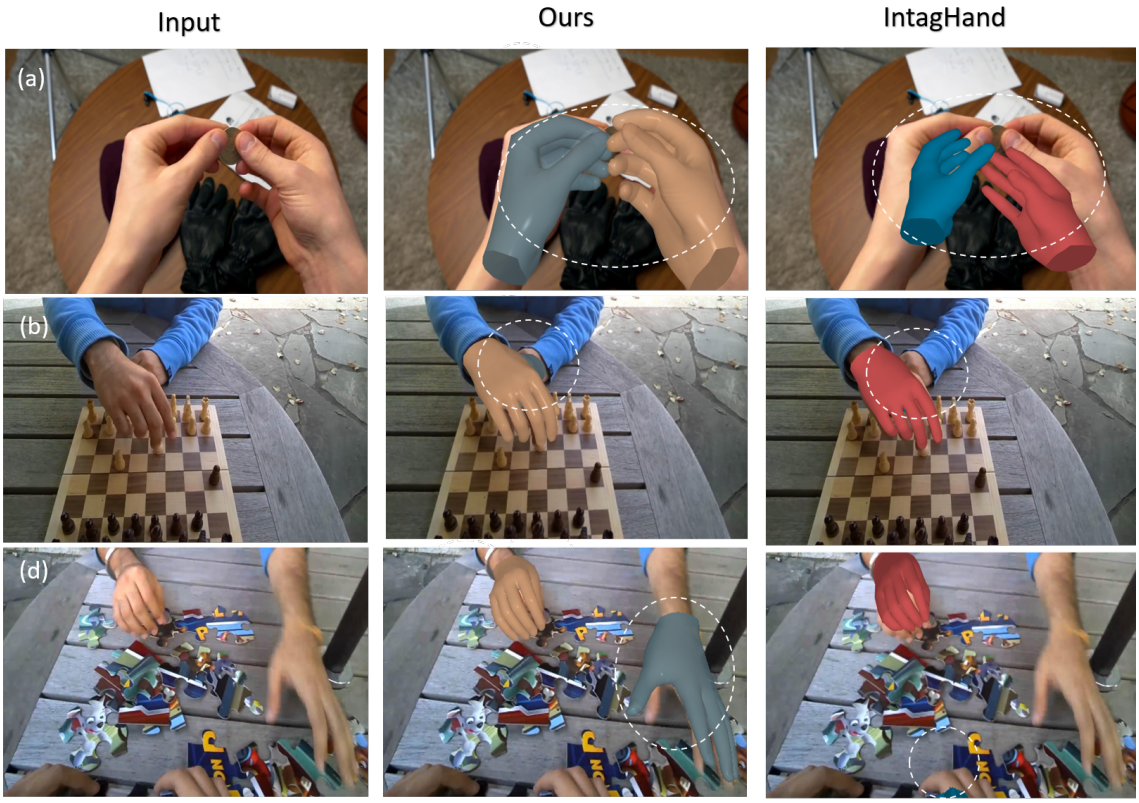


Figure 4.8: Qualitative comparison results on ego-view data. images in (b)(c)(d) are selected from RGB2Hands benchmark[9].

is very sensitive to external occlusion, as it may treat the object occlusion as interacting hand occlusion, resulting in failure estimation. More interestingly, IntagHand mostly fails to reconstruct two hands on the ego-view dataset (as shown in Figure 4.8). One possible reason is its GCN and transformer-based attention mechanism overly rely on two-hand interacting dependency to reason about two-hand reconstruction. At the same time, the two hands are primarily separate and coupled with slight object occlusion. Nevertheless, our method consistently performs well on this dataset thanks to its independent features for each hand and its powerful collaborative representation.

4.5.4 Real-time and In-the-wild Applications

We implement a real-time two-hand reconstruction demo based on our proposed method and an ordinary webcam. Due to simplicity, our approach can run in real-

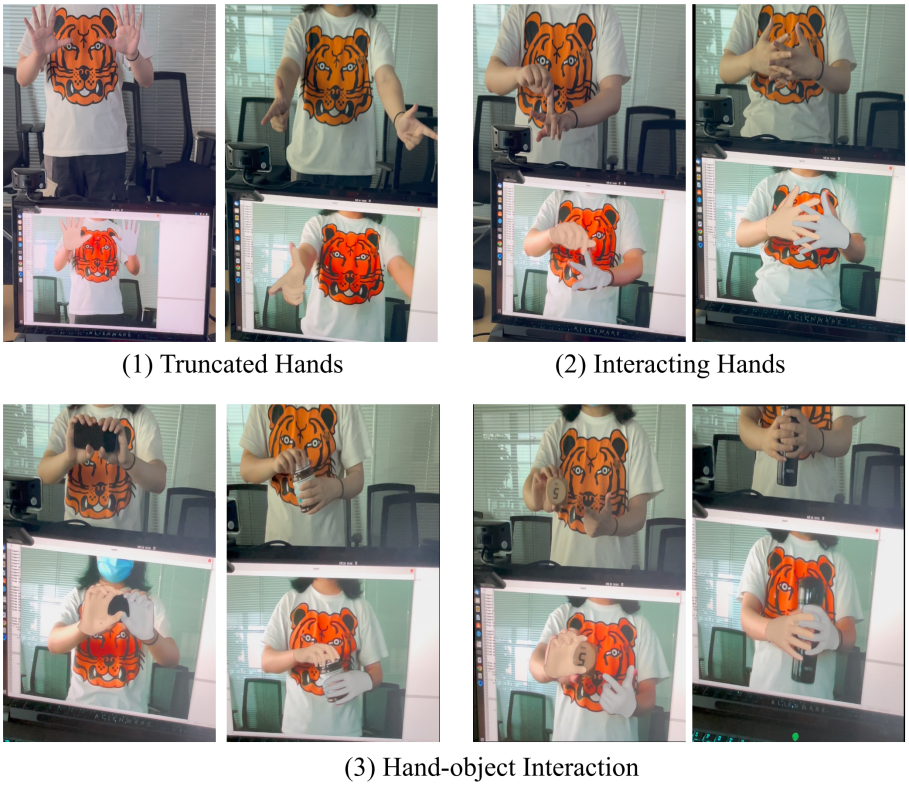


Figure 4.9: Results of the ACR real-time demo (see video ***acr_live_demo.mp4*** for more detail.) Our method produces high-quality results on a live video stream from a cheap webcam.

time on a laptop with an RTX 2080 GPU. We provide the results of the demo in Figure 4.9 and a video ***acr_live_demo.mp4*** at this link for more details. Our method can produce high-quality reconstruction results and effectively handle various inputs such as interacting hands, truncated hands, and hand-object interaction. Besides, our algorithm bypasses the requirement of a hand detector or constraint inputs, while IntagHand [8] requires two-hand in a pre-defined region. These advantages are significant for advancing hand-reconstruction technology in real-world applications. Moreover, due to this high flexibility and freedom, we can easily perform in-the-wild applications on random videos and they can be found at this link.

4.5.5 Ablation study

As introduced in Section 4.3, our Attention Collaboration-based Feature Aggregator (ACFA) works mainly by collaborating three representations: global representation (G, baseline), part-based representation (P), and cross-hand attention prior (C).

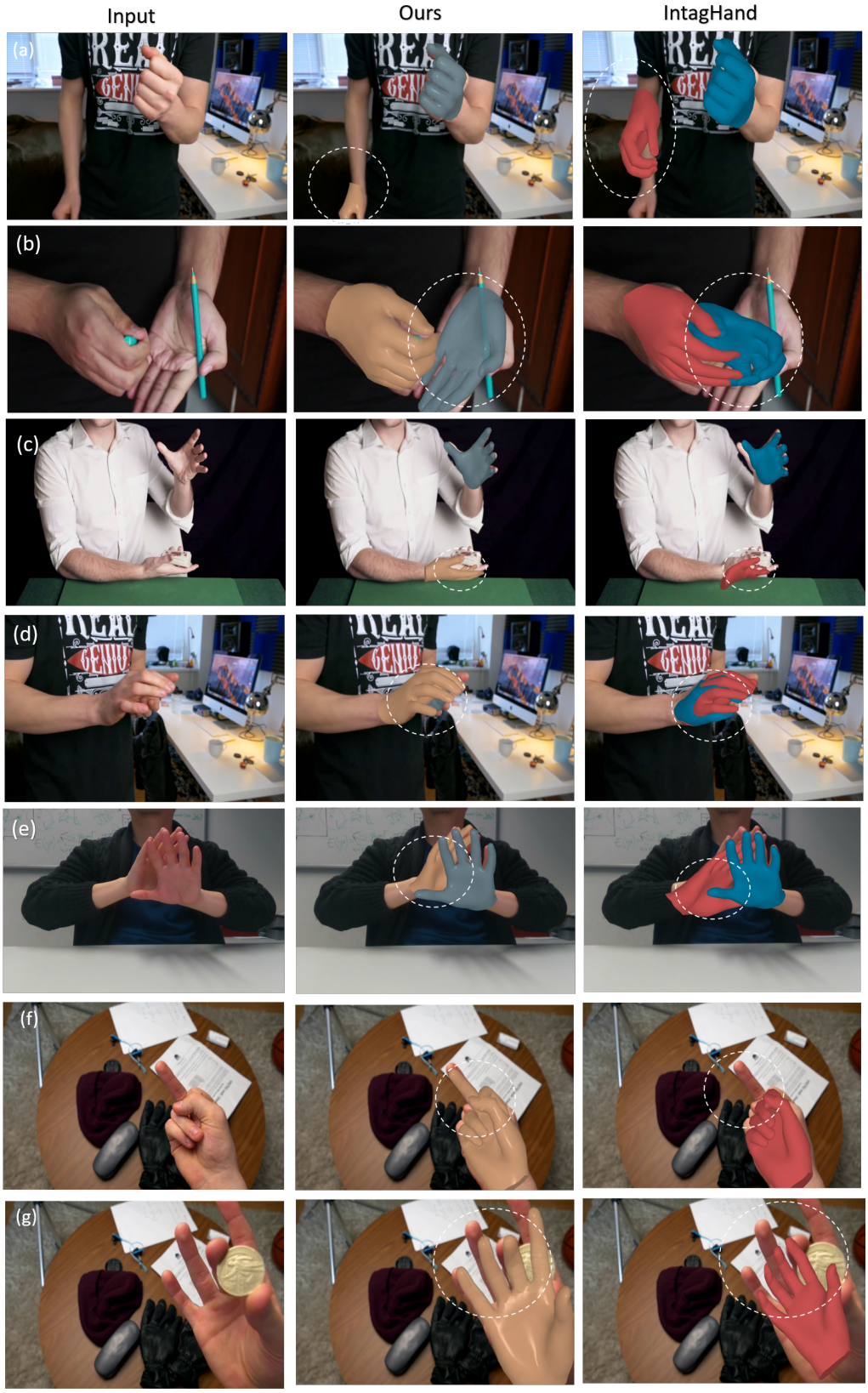


Figure 4.10: Interacting hand and single hand reconstruction. Here, the images in (e) are selected from RGB2Hands benchmark[9]. The others are from web videos.

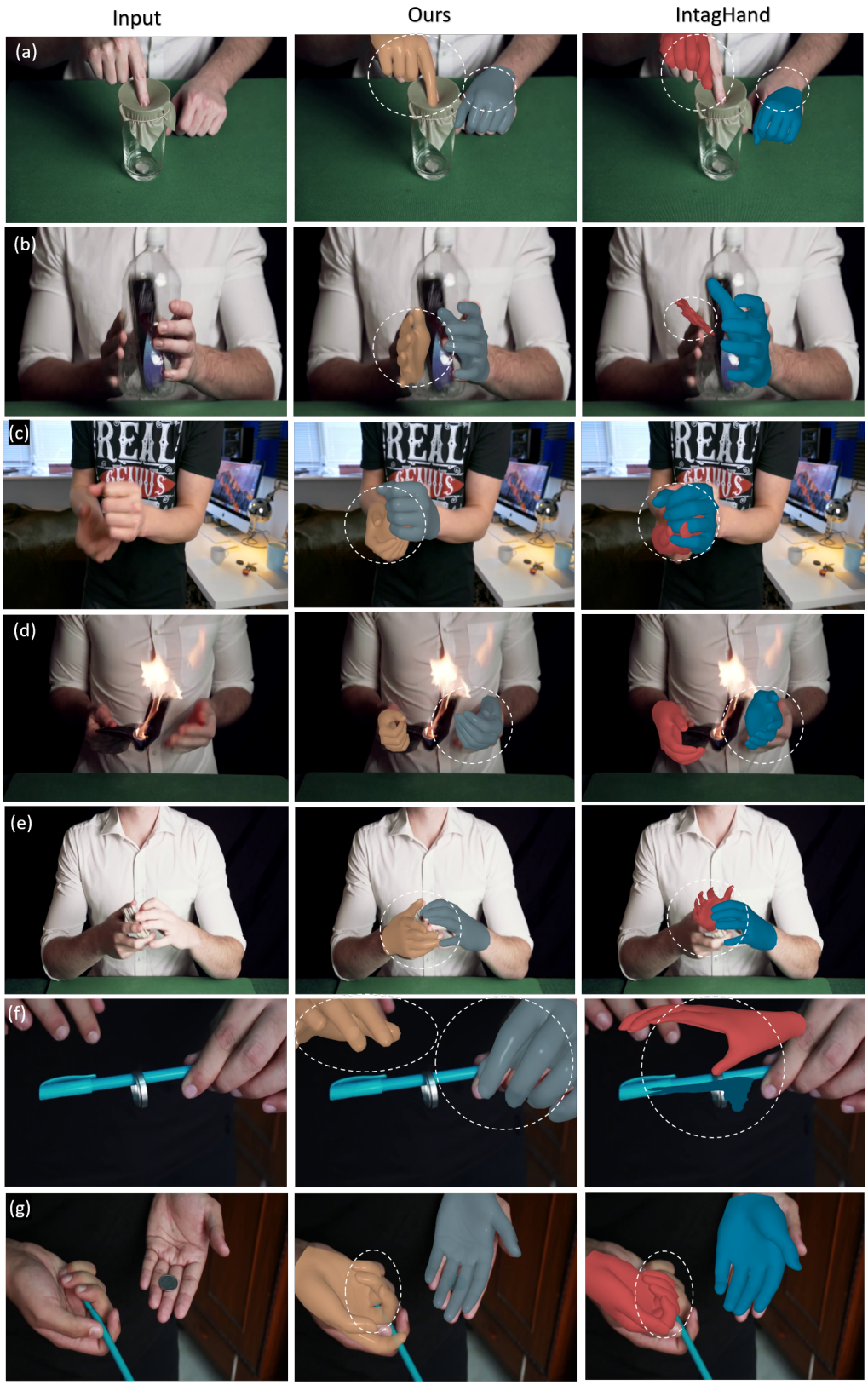


Figure 4.11: Hand-object interaction on web videos (watch video ***acr_in_the_wild.mp4*** for more detail).

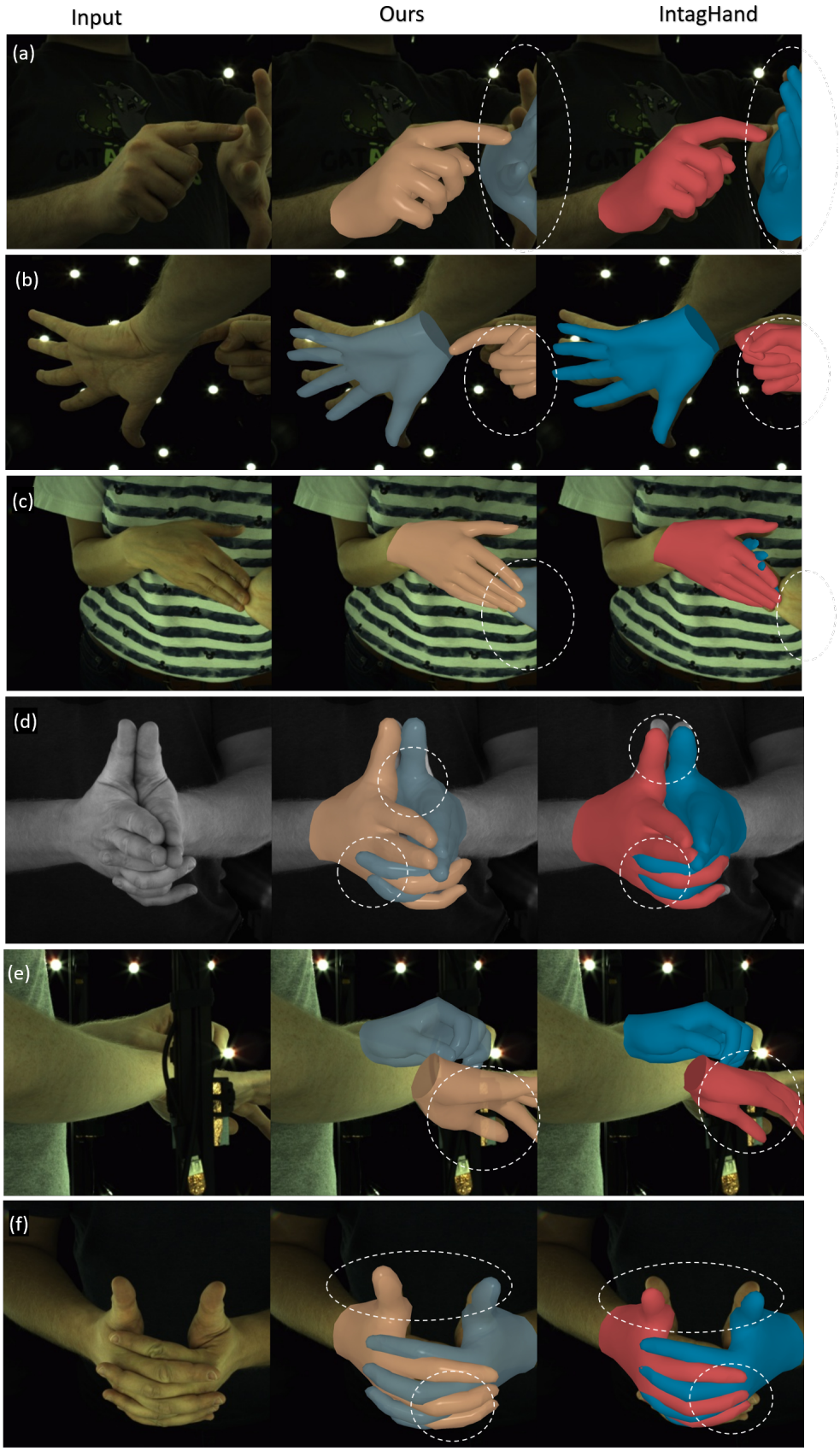


Figure 4.12: Extra qualitative results for InterHand2.6M dataset.

	MPJPE	IH MPJPE	SH MPJPE	PAMPJPE
G(ResNet-50)	9.78	10.56	8.77	6.56
G(HRNet-32W)	9.56	10.35	8.65	6.41
P	8.70	9.76	7.26	5.59
G+C	9.1	9.88	8.11	6.08
G+P	8.52	9.69	6.87	5.49
G+**C**+P	**8.09**	**9.08**	**6.85**	**5.21**

Table 4.3: Ablation study on the part (P), global (G), and cross-hand (C) prior representation. We do not use any extra information such as the bounding box and GT scale in the ablation study.

Therefore, we investigate the effectiveness of each module. We treat the center-based representation as the baseline and gradually add other modules to see their improvements. As shown in Table 4.3, we can clearly observe both part-based and cross-hand significantly improve the baseline. More interestingly, the improvement of adding C on the IH dataset is more significant than that on the SH dataset. This demonstrates that the cross-hand attention prior facilitates addressing interacting hand challenges. We also provide further information about our network and ablation studies for the aggregation method and supervision method as below.

Ablation study of aggregation method To explore a proper way to aggregate the global representation and part representation while maintaining their own advantages, we have conducted different kinds of aggregation methods (*mode* in the Tab 4.4), where *offset* means a simple summation and *concat* means feature aggregation illustrated in Section 4.3. We found that aggregating the representations by concatenation always yields better performance under different cases. As a result, we report the final results and claim the state-of-the-art in this manner.

Ablation study of supervision: In Tab 4.4, in addition to decoupling each module of our network, we also explored 1) different ways to aggregate the global representation and part-based representation. We first tried to remove the supervision of L_{seg} from our network to see if the part segmentation can work as implicit attention guidance, and vice versa, if the part segmentation can explicitly work as an attention mask. Finally, we use a hybrid training strategy for our network, which only supervise the L_{seg} for the first two epoch. This strategy significantly speeds

	mode	supervision	IH MPJPE	SH MPJPE	PAMPJPE
G	-	-	10.35	8.65	6.41
G+P	Concat	Full	9.71	7.05	5.54
P	-	Full	10.03	7.48	5.68
G+P	Offset	Full	9.82	7.13	5.60
G+P	Concat	Hybrid	9.69	6.87	5.49
P	-	Hybrid	9.76	7.26	5.59
G+P	Offset	Hybrid	9.49	6.91	5.50
G+P	Concat	Unsup.	9.73	7.05	5.54
P	-	Unsup.	10.05	7.52	5.67
G+P	Offset	Unsup.	9.87	7.17	5.61
G+**C**+P	Offset	Hybrid	9.28	7.01	5.38
G+**C**+P	Concat	Hybrid	**9.08**	**6.85**	**5.21**

Table 4.4: Ablation study of different aggregation methods of part-global representation learning, cross-hand-attention prior module, and part-segmentation branch supervision method. *mode* means the aggregation method of part-global representation. *supervision* suggests different supervision strategies for the art segmentation branch. G, P, and C stand separately for global representation, part-based representation, and cross-hand attention prior module.

up the training process and yields the best performance. Please note that having a superior segmentation mask is unnecessary to learn a part-based representation. On the contrary, it is reasonable to have a slightly lower mIoU, as this would expand the attention area of each part to let the network focus on the visible parts and aggregate helpful features for the missing part by a reasonable deduction.

4.6 Discussion and Conclusion

Conclusion: This work attempts to address a more challenging problem that reconstructs arbitrary hand poses and shapes from a single RGB image. We present a simple yet effective approach considering more challenges such as interacting hands, truncated hands, and external occlusion and separation, while existing approaches [8, 14, 57] all adopt an entangled representation for hand recovery, which can only deal with either single hand or **closely** interacting hands by combining with an external detector as introduced in Section 2.1 of Chapter 2. To this end, we propose to leverage center and part attention to mitigate interdependencies between hands and between parts to release the input constraint and eliminate the predictions sensitivity

to a small occluded or truncated part. Moreover, we propose to explicitly represent the interaction intensity with an adaptive interaction field. Benefiting from the design of representation disentanglement and attention collaboration, experiments show that our method achieves the state-of-the-art on the existing interacting hand dataset. Furthermore, our method is the most practical method without the need of an external detector, which can serve as a baseline to inspire more research on arbitrary hand pose and shape reconstruction and AR or VR applications.

Limitation and Future Work: Our major limitation is the lack of an explicit solution for physical mesh collision, resulting in occasional inter-penetration, which could also be solved by leveraging relative information or perspective camera model for accurate depth reasoning and better simulation of translation to some extend.

Discussion: In terms of monocular full-body capturing, hand pose estimation is always the most difficult part due to frequent occlusion, truncation, and fast movements. Typical methods follow a pipeline to crop the single hand by an external hand detector. However, it has been explored that interacting hands can not be well recovered separately by treating them as single hand and applying reconstruction methods to them individually. Thus, all the existing interacting hand reconstruction methods naturally adopt a naive strategy to crop the interacting two hands in one box and extend the output to two hands with some tailored mutual fusion part such as transformer-based [8] module. We believe this kind of cropping strategy is an inherently ill-posed pipeline because it is only limited to very closely interacting hands without generalization ability. On the contrary, our method is the only one that can be plugged into any kind of hand pose estimation task for arbitrary hand reconstruction. It is also worth noting that our network can be very easy to be extended to the multiple-hand setting by leveraging the center representation.

CHAPTER 5

Conclusions

5.1 Summary of Key Contributions

Overall, this thesis makes the first steps towards building up an intelligent system that can recover, understand geometry and semantics, and finally interact with the real world. Specifically, we explore explicit to implicit representation methods on both scenes and humans.

In Chapter 3, we introduce our RangeUDF, a new implicit representation method to recover the geometry and semantics of continuous 3D scene surfaces from sparse raw 3D point clouds. Our method is the first one to directly reconstruct 3D semantic surface from sparse point clouds. To enable efficient learning from large-scale sparse raw point clouds, we leverage a point-wise encoder to encode the sparse surfaces into the implicit function. With the continuous, encoded implicit function, the feature vectors of query points in the empty space is then formulated by coupling with the K nearest neighbours on the sparse surface. Existing approaches [30, 81–84] typically adopt voxel-based encoding and feature query with a trilinear interpolation that suffers from the surface ambiguity problem and results in over-smooth surfaces and loss of details as introduced in Section 2.2 of Chapter 2. In contrast, our method

no longer suffers from the surface ambiguity brought by trilinear interpolation and therefore can produce more fine-grained 3D surfaces by leveraging the range-aware neural interpolation module for unsigned distance function learning. Moreover, it is hard for voxelization-based methods such as NDF [6] to integrate semantic information for further application due to the reliance on a slow and inappropriate sliding window approach to process the whole scene. In contrast, our method directly reconstructs accurate 3D semantic surfaces by coupling the reconstruction with a surface-oriented semantic segmentation module to infer semantic classifications for the underlying implicit surface. Benefiting from these designs, our method has been shown to outperform the previous state-of-the-art methods [6, 30] by a large margin with much less computational and time consumption. In addition, our work is the first one that has superior generalization ability in cross-domain settings and bridges the gap between real-world data and synthetic data.

In Chapter 4, we explore explicit representation in a new trending direction of AR and VR, parametric 3D reconstruction of human hands. We present ACR in this chapter, an attention collaboration-based regressor, which makes the first attempt to reconstruct hands in arbitrary scenarios. In contrast to all the existing hand mesh reconstruction approaches such as [8, 13, 14, 57, 113], our method is the first one-stage method directly from raw RGB image input without the reliance on an external hand detector. As introduced in Section 2.1 of Chapter 2, existing two hand reconstruction methods [8, 13, 14, 113] all suffer from the entangled bounding-box-level features by treating the two hands as an integral to learn an entangled representation for two hands simultaneously, which are incredibly fragile to impaired interaction, such as truncated hands, separate hands, or external occlusion. In contrast, our method explicitly mitigates the inter-dependencies between two hands and parts of each hand by disentangling the representation into center-based representation and part-based representation to unleash the power of point-wise representation. However, we believe that the two interacting hands are highly correlated, and disentangling features of the two hands will also weaken the mutual reasoning ability to model interacting hands. Therefore, we design a cross-hand prior learning module based on the center attention together with interaction ten-

sity aware fields for adjusting the dependency strength dynamically. Benefiting from these designs, extensive experimental results on the existing benchmarks [9, 13, 15] and in-the-wild data demonstrate the superiority of our method against previous state-of-the-art approach [8]. Moreover, our method is the first one-stage method in hand reconstruction area, and thus is the most compatible one for real-world applications of full-body motion capture leveraging the powerful representation with the most promising results in this area so far and the highest degree of freedom without the need of any external information such as detector.

5.2 Limitations and Future Works

Although having superior state-of-the-art results, the works in this thesis also open up many potential research directions for future research work:

Photorealistic Holistic Scene Reconstruction: Despite the promising results that representations proposed in the thesis have achieved, they are static representations and agnostic to the time dimension. Exploring another pipeline of implicit representation - NeRF series [23] will be a wonderful choice for continuation and pursuing the aim of dynamic 3D representation.

Meshing Algorithms for Implicit Representations: In chapter 3, we propose an implicit representation learning based method for scene reconstruction understanding. However, we still rely on the off-the-shelf Marching Cubes algorithm to extract the mesh from the dense generated point clouds.

Scene Decomposition and Editing: As a crucial technique for applications in augmented reality, scene decomposition and disentanglement have also been studied recently on both static representation of point cloud data [125] and dynamic representation of neural radiance fields [126].

3D Scene Capture and Generation with Generative Models: We can reconstruct/generate different realizations or layouts of dynamic indoor scenes from sparse inputs. Some very recent works had found the diffusion model [127, 128] very powerful in human motion generation as well. Since the diffusion model can also be used to generate temporal images of the same object in different states and times;

We can also utilize it for efficiently capturing dynamic scene reconstruction from sparse views with inpainting or completion online or novel view synthesis.

Another more challenging task that no one has done before could be: dynamic 3D indoor scene generation with conditioning semantic guidance/text (e.g., raw semantic segmentation, or text descriptions like *"A wooden table on the ground with a glass cup on it"*) Due to the lack of 3D content datasets, we maybe can explore a way to use 2D diffusions priors [129, 130]. We believe the predefined structural information can further improve the quality of the generated 3D content and make it more photorealistic and diverse.

Human-scene Interaction: As the largest intersection over these research realms, it serves as a key to connecting people with the digital world and modelling interactions in the digital world can serve many purposes. However, this requires both high-quality dynamic 3D representation of scenes and humans, which can be a future aim after achieving promising results in the above directions.

In conclusion, this thesis takes a step towards the challenge of building up the intelligent system to recover, understand and interact with the real world by exploring from explicit representation to implicit representation on scene and humans. We hope this could be an insight in these directions for researchers to go further and present more solid amazing works, which will the key for people to connect the digital world with our real world together with the help of artificial intelligence with various applications.

Bibliography

[1] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015. viii, 7, 44

[2] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 10975–10985, 2019. 8, 44

[3] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics*, vol. 36, no. 6, 2017. viii, 5, 7, 48, 50

[4] S. Zuffi, A. Kanazawa, D. Jacobs, and M. J. Black, "3D menagerie: Modeling the 3D shape and pose of animals," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. viii, 7

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. viii, ix, xii, xiii, 2, 14, 28, 33, 37, 38, 40, 41, 42, 87

[6] J. Chibane, G. Pons-Moll, *et al.*, "Neural unsigned distance fields for implicit function learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21638–21652, 2020. viii, 3, 5, 9, 10, 11, 14, 15, 16, 20, 21, 26, 27, 28, 29, 30, 33, 34, 36, 42, 73

[7] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 11108–11117, 2020. ix, 10, 18, 19, 36

[8] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *IEEE/CVF Conf.*

on Computer Vision and Pattern Recognition (CVPR), June 2022. ix, x, 7, 8, 11, 45, 46, 49, 51, 53, 59, 61, 62, 63, 65, 70, 71, 73, 74

[9] J. Wang, F. Mueller, F. Bernard, S. Sorli, O. Sotnychenko, N. Qian, M. A. Otaduy, D. Casas, and C. Theobalt, "Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video," *ACM Transactions on Graphics (ToG)*, vol. 39, no. 6, pp. 1–16, 2020. x, 8, 46, 60, 63, 64, 66, 74

[10] Z. Yu and S. Huang, "Towards accurate alignment for representation learning of human pose estimation in-the-wild," in *Tencent AI Lab 2022*. `https://sites.google.com/view/zhengdiyu/projects?authuser=0#h.jhdudhm4mh6`. xi, 89

[11] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "Scenenn: A scene meshes dataset with annotations," in *2016 fourth Int. Conf. 3D vision (3DV)*, pp. 92–101, Ieee, 2016. xiii, 2, 28, 29, 33, 38, 40, 42, 87, 88

[12] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017. xiii, xiv, 2, 28, 33, 38, 40, 42, 87, 88

[13] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *European Conference on Computer Vision (ECCV)*, 2020. xiii, 2, 47, 51, 60, 61, 73, 74

[14] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 11354–11363, 2021. xiii, 8, 11, 46, 51, 53, 59, 61, 70, 73

[15] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 813–822, 2019. xiii, 60, 62, 74

[16] O. zyeil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion.," *Acta Numerica*, vol. 26, p. 305364, 2017. 1, 6

[17] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 4104–4113, 2016. 1, 6

[18] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, pp. 1309–1332, dec 2016. 1, 6

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015. 1, 6

[20] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015. 2, 28, 29

[21] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 9297–9307, 2019. 2

[22] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *Proc. European Conf. Computer Vision (ECCV)*, pp. 601–617, 2018. 2

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021. 3, 9, 15, 74

[24] B. Wang, C. Chen, Z. Cui, J. Qin, C. X. Lu, Z. Yu, P. Zhao, Z. Dong, F. Zhu, N. Trigoni, *et al.*, "P2-net: Joint description and detection of local features for pixel and point matching," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 16004–16013, 2021. 4

[25] W. Bing, Y. Zhengdi, B. Yang, Q. Jie, B. Toby, S. Ling, N. Trigoni, and A. Markham, "Rangeudf: Semantic surface reconstruction from 3d point clouds," *arXiv preprint arXiv:2204.09138*, 2022. 4, 13

[26] Z. Yu, S. Huang, C. Fang, T. Breckon, and J. Wang, "Acr: Attention collaboration-based regressor for arbitrary two-hand reconstruction," in *submission to CVPR*, 2023. `https://sites.google.com/view/zhengdiyu/publications#h.me7vds3d4g22`. 4, 7, 44

[27] Z. Yu, "Ailabmocap: Full-body 3d pose estimation and motion capture system," in *Tencent AI Lab Key Program*, 2022. `https://sites.google.com/view/zhengdiyu/projects#h.v7vdcsgoa79k`. 4

[28] Epic Games, "Unreal engine." `https://www.unrealengine.com`. 4

[29] B. O. Community, *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. `http://www.blender.org`. 4

[30] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conf. Computer Vision*, pp. 523–540, Springer, 2020. 5, 10, 11, 15, 20, 28, 29, 30, 33, 34, 36, 42, 72, 73

[31] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, vol. 36, pp. 301–329, Wiley Online Library, 2017. 6

[32] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Eur. Conf. on Computer Vision*, pp. 628–644, Springer, 2016. 7

[33] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proc. IEEE Int. Conf. computer vision*, pp. 2088–2096, 2017. 7

[34] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 605–613, 2017. 7, 30

[35] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 3907–3916, 2018. 7, 51, 58, 59

[36] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "Pare: Part attention regressor for 3d human body estimation," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 11127–11137, 2021. 7, 49, 52

[37] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem, "3d-prnn: Generating shape primitives with recurrent neural networks," in *Proc. IEEE Int. Conf. Computer Vision*, pp. 900–909, 2017. 7

[38] G. Gkioxari, J. Malik, and J. Johnson, "Mesh r-cnn," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 9785–9795, 2019. 7

[39] S. Tulsiani, S. Gupta, D. F. Fouhey, A. A. Efros, and J. Malik, "Factoring shape, pose, and layout from the 2d image of a 3d scene," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 302–310, 2018. 7

[40] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *Proc. IEEE/CVF Int. Conf. computer vision*, pp. 2690–2698, 2019. 7

[41] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen, "Dense 3d object reconstruction from a single depth view," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2820–2834, 2018. 7

[42] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 1746–1754, 2017. 7

[43] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3d object reconstruction," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 32, 2018. 7

[44] H. Choi, G. Moon, J. Park, and K. M. Lee, "Learning to estimate robust 3d human mesh from in-the-wild crowded scenes," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 1475–1484, 2022. 7

[45] M. Kocabas, C.-H. P. Huang, J. Tesch, L. Müller, O. Hilliges, and M. J. Black, "Spec: Seeing people in the wild with an estimated camera," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 11035–11045, 2021.

[46] A. Sengupta, I. Budvytis, and R. Cipolla, "Hierarchical kinematic probability distributions for 3d human shape and pose estimation from images in the wild," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 11219–11229, 2021.

[47] A. Sengupta, I. Budvytis, and R. Cipolla, "Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 16094–16104, 2021. 7

[48] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose," in *European Conf. Computer Vision*, pp. 769–787, Springer, 2020. 7

[49] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 7122–7131, 2018. 7

[50] J. N. Kundu, M. Rakesh, V. Jampani, R. M. Venkatesh, and R. Venkatesh Babu, "Appearance consensus driven self-supervised human mesh recovery," in *European Conf. Computer Vision*, pp. 794–812, Springer, 2020. 7

[51] Y. Xu, S.-C. Zhu, and T. Tung, "Denserac: Joint 3d pose and shape estimation by dense render-and-compare," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 7760–7770, 2019. 7

[52] Y. Sun, Y. Ye, W. Liu, W. Gao, Y. Fu, and T. Mei, "Human mesh recovery from monocular images via a skeleton-disentangled representation," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 5349–5358, 2019. 7

[53] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 5253–5263, 2020. 7

[54] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 5614–5623, 2019. 7

[55] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *European Conf. computer vision*, pp. 561–578, Springer, 2016. 7

[56] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3d human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 2252–2261, 2019. 7

[57] A. Boukhayma, R. d. Bem, and P. H. Torr, "3d hand shape and pose from images in the wild," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 10843–10852, 2019. 7, 46, 49, 61, 70, 73

[58] Z. Baowen, W. Yangang, D. Xiaoming, Z. Yinda, T. Ping, M. Cuixia, and W. Hongan, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Int. Conf. Computer Vision (ICCV)*, 2021. 7

[59] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *arXiv preprint arXiv:2201.02610*, 2022. 7

[60] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 7291–7299, 2017. 8

[61] H. Joo, T. Simon, and Y. Sheikh, "Total capture: A 3d deformation model for tracking faces, hands, and bodies," in *Proc. the IEEE conference on computer vision and pattern recognition*, pp. 8320–8329, 2018. 8

[62] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proc. the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10974, 2019.

[63] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *European Conference on Computer Vision*, pp. 20–40, Springer, 2020.

[64] Y. Zhang, Z. Li, L. An, M. Li, T. Yu, and Y. Liu, "Lightweight multi-person total motion capture using sparse multi-view cameras," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 5560–5569, 2021.

[65] Y. Zhou, M. Habermann, I. Habibie, A. Tewari, C. Theobalt, and F. Xu, "Monocular real-time full body capture with inter-part correlations," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4811–4822, 2021.

[66] Y. Feng, V. Choutas, T. Bolkart, D. Tzionas, and M. J. Black, "Collaborative regression of expressive bodies using moderation," in *2021 Int. Conf. 3D Vision (3DV)*, pp. 792–804, IEEE, 2021. 8

[67] J. Taylor, V. Tankovich, D. Tang, C. Keskin, D. Kim, P. Davidson, A. Kowdle, and S. Izadi, "Articulated distance fields for ultra-fast tracking of hands interacting," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–12, 2017. 8

[68] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt, "Real-time pose and shape reconstruction of two interacting hands with a single depth camera," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019. 8

[69] B. Smith, C. Wu, H. Wen, P. Peluse, Y. Sheikh, J. K. Hodgins, and T. Shiratori, "Constraining dense hand surface tracking with elasticity," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–14, 2020. 8

[70] Y. Rong, J. Wang, Z. Liu, and C. C. Loy, "Monocular 3d reconstruction of interacting hands via collision-aware factorized refinements," in *2021 Int. Conf. 3D Vision (3DV)*, pp. 432–441, IEEE, 2021. 8, 46

[71] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019. 9

[72] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 4460–4470, 2019. 9, 15, 30, 33

[73] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 9

[74] M. Atzmon and Y. Lipman, "Sal: Sign agnostic learning of shapes from raw data," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 2565–2574, 2020. 9, 15

[75] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021. 9

[76] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3d shape reconstruction and completion," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6970–6981, 2020. 9, 15

[77] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 6229–6238, 2021. 9, 15

[78] A. Luo, T. Li, W.-H. Zhang, and T. S. Lee, "Surfgen: Adversarial 3d shape synthesis with explicit surface discriminators," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 16238–16248, 2021. 9

[79] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020. 9

[80] C. Zhang, Z. Cui, Y. Zhang, B. Zeng, M. Pollefeys, and S. Liu, "Holistic 3d scene understanding from a single image with implicit representation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 8833–8842, 2021. 9, 15

[81] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 165–174, 2019. 10, 15, 72

[82] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, "Deep local shapes: Learning local sdf priors for detailed 3d reconstruction," in *European Conf. Computer Vision*, pp. 608–625, Springer, 2020.

[83] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser, *et al.*, "Local implicit grid representations for 3d scenes," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 6001–6010, 2020.

[84] S. Lombardi, M. R. Oswald, and M. Pollefeys, "Scalable point cloud-based reconstruction with local implicit functions," in *2020 Int. Conf. 3D Vision (3DV)*, pp. 997–1007, IEEE, 2020. 10, 72

[85] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987. 10, 27

[86] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 9224–9232, 2018. 10

[87] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 652–660, 2017. 10, 36

[88] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 9404–9413, 2019. 10

[89] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 6504–6513, 2021. 11, 28, 34, 36

[90] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013. 15

[91] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 5589–5599, 2021. 15

[92] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021. 15

[93] A. Trevithick and B. Yang, "Grf: Learning a general radiance field for 3d representation and rendering," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 15182–15192, 2021. 15

[94] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. computer vision*, pp. 6411–6420, 2019. 19, 36, 40

[95] B. Yang, S. Wang, A. Markham, and N. Trigoni, "Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction," *Int. J. of Computer Vision*, vol. 128, no. 1, pp. 53–73, 2020. 22, 24

[96] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 22

[97] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. computer vision and pattern recognition*, pp. 7482–7491, 2018. 25

[98] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019. 25, 57

[99] F.-E. Wolter, "Cut locus and medial axis in global shape interrogation and representation," 1993. 26

[100] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999. 27

[101] B. Ma, Z. Han, Y.-S. Liu, and M. Zwicker, "Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces," *arXiv preprint arXiv:2011.13495*, 2020. 28, 33

[102] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017. 30

[103] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?," in *Proc. IEEE/CVF Conf. computer vision and pattern recognition*, pp. 3405–3414, 2019. 30

[104] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll, "Video based reconstruction of 3d people models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 8387–8397, 2018. 44

[105] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor, "Tex2shape: Detailed full human body geometry from a single image," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, pp. 2293–2303, 2019.

[106] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll, "Multi-garment net: Learning to dress 3d people from images," in *Proc. IEEE/CVF Int. Conf. computer vision*, pp. 5420–5430, 2019. 44

[107] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng, "End-to-end hand mesh recovery from a monocular rgb image," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 2354–2364, 2019. 46

[108] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1067–1076, 2019.

[109] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu, "Monocular real-time hand shape and motion capture using multi-modal data," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5346–5355, 2020. 46, 61

[110] X. Tang, T. Wang, and C.-W. Fu, "Towards accurate alignment in real-time 3d hand-mesh reconstruction," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 11698–11707, 2021. 46

[111] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2019. 48

[112] Y. Sun, Q. Bao, W. Liu, Y. Fu, B. Michael J., and T. Mei, "Monocular, one-stage, regression of multiple 3d people," in *ICCV*, 2021. 49, 52

[113] Z. Fan, A. Spurr, M. Kocabas, S. Tang, M. Black, and O. Hilliges, "Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation," in *Int. Conf. 3D Vision (3DV)*, 2021. 49, 58, 61, 73

[114] J. Lv, W. Xu, L. Yang, S. Qian, C. Mao, and C. Lu, "Handtailor: Towards high-precision monocular 3d hand recovery," *arXiv preprint arXiv:2102.09244*, 2021. 62

[115] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," tech. rep., arXiv:1705.01389, 2017. https://arxiv.org/abs/1705.01389. 49, 61

[116] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. the IEEE Int. Conf. computer vision*, pp. 2980–2988, 2017. 55

[117] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 57

[118] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in

Proc. the IEEE/CVF conference on computer vision and pattern recognition, pp. 5386–5395, 2020. 57

[119] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," *Advances in neural information processing systems*, vol. 31, 2018. 57

[120] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. the IEEE Int. Conf. computer vision*, pp. 1949–1957, 2015. 60, 63

[121] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–98, 2018. 61

[122] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proc. the IEEE/CVF Int. Conf. Computer Vision*, pp. 12939–12948, 2021. 62

[123] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021. 62

[124] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image," in *European Conference on Computer Vision*, pp. 752–768, Springer, 2020. 62

[125] V. Guzov, T. Sattler, and G. Pons-Moll, "Visually plausible human-object interaction capture from wearable sensors," *arXiv preprint arXiv:2205.02830*, 2022. 74

[126] B. Wang, L. Chen, and B. Yang, "Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images," *arXiv preprint arXiv:2208.07227*, 2022. 74

[127] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Int. Conf. Machine Learning*, pp. 2256–2265, PMLR, 2015. 74

[128] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020. 74

[129] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022. 75

[130] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," *arXiv preprint arXiv:2211.10440*, 2022. 75

APPENDIX A

Real-world Applications and Auxiliary Results

A.1 More results of RangeUDF

In this section, we provide per-class evaluation results of RangeUDF on three real-world datasets: ScanNet [5], SceneNN [11] and 2D-3D-S [12]. Please note that it is not fair to directly compare our method to normal semantic segmentation methods because their evaluation is conducted on the discrete point clouds while our method infers semantics for the implicit surface.

Methods	**OA (%)**	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	refrigerator	shower curtain	toilet	sink	bathtub	other furniture
Ours	83.2	0.88	0.99	0.65	0.82	0.87	0.84	0.78	0.57	0.59	0.69	0.11	0.63	0.60	0.69	0.56	0.44	0.76	0.60	0.78	0.34

Table A.1: RangeUDF: Semantic accuracy on ScanNet [5]

Methods	**mIoU (%)**	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	refrigerator	shower curtain	toilet	sink	bathtub	other furniture
Ours	0.55	0.76	0.95	0.49	0.67	0.77	0.71	0.63	0.39	0.41	0.54	0.10	0.47	0.44	0.55	0.38	0.39	0.74	0.53	0.72	0.28

Table A.2: RangeUDF: Semantic mIoU on ScanNet [5]

Methods	**OA (%)**	wall	floor	cabinet	bed	chair	sofa	table	desk	tv	box	props
Ours	0.86	0.97	0.93	0.25	0.80	0.76	0.60	0.49	0.63	0.56	0.40	0.20

Table A.3: RangeUDF: Semantic accuracy on SceneNN [11]

Methods	**mIoU (%)**	wall	floor	cabinet	bed	chair	sofa	table	desk	tv	box	props
Ours	0.46	0.90	0.91	0.16	0.57	0.59	0.17	0.34	0.48	0.49	0.31	0.15

Table A.4: RangeUDF: Semantic mIoU on SceneNN [11]

Methods	**OA (%)**	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Ours	0.89	0.99	0.98	0.91	0.83	0.78	0.77	0.87	0.80	0.87	0.89	0.83	0.73	0.75

Table A.5: RangeUDF: Semantic accuracy on 2D-3D-S [12]

Methods	**mIoU (%)**	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Ours	0.74	0.96	0.95	0.82	0.79	0.61	0.67	0.73	0.70	0.81	0.73	0.64	0.63	0.62

Table A.6: RangeUDF: Semantic mIoU on 2D-3D-S [12]

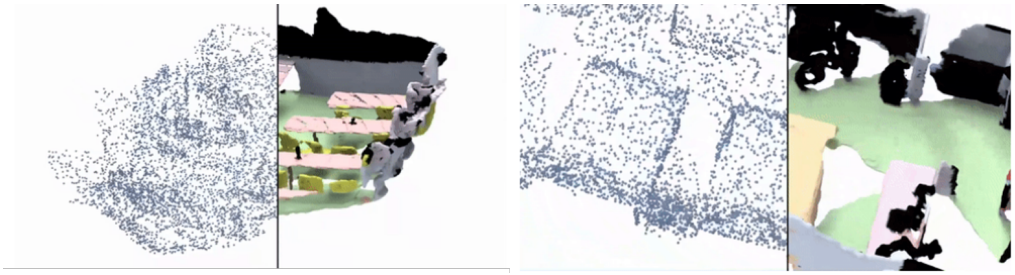


Figure A.1: **RangeUDF demos:** The scenes split by the black line. The left side is the raw point cloud of the area. Full videos can be found at: this link

A.2 Other Demos

In this section, some demos are presented for the body reconstruction and motion capture system. Please find the full videos in the link below the image or the supplementary videos on my personal website.

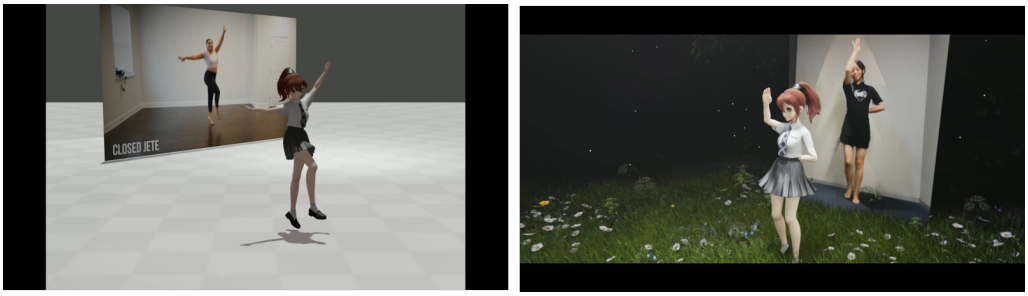


Figure A.2: **Blender demos:** Character driven by the aforementioned paper ACR in chapter 4 and the project [10], which is not included in the main chapters. Please found videos at this link and thins link

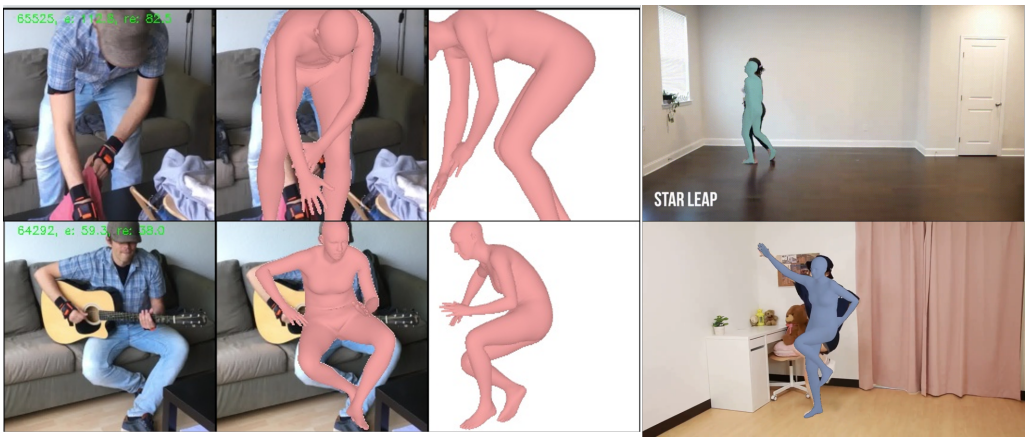


Figure A.3: **Mesh rendering results** of the project [10], which is not included in the main chapters.