Thesis

Improving image-based 3D Human Mesh Recovery with LiDAR data

MSc Robotics
Guido Dumont

TU Delft
2024

Master of Science Robotics

Thesis research

Improving image-based 3D Human Mesh Recovery with LiDAR data

by

Guido Dumont

Supervisors:

Dr. Javier Also Mora - Main supervisor

Clarence Chen - Daily supervisor

Date: July 10, 2024

Abstract

Human Mesh Recovery (HMR) frameworks predict a comprehensive 3D mesh of an observed human based on sensor measurements. The majority of these frameworks are purely image-based. Despite the richness of this data, image-based HMR frameworks are vulnerable to depth ambiguity, resulting in mesh inaccuracies in 3D space. Several HMR frameworks in the literature use LiDAR data to avoid this depth ambiguity. However, these frameworks are either only LiDAR-based, which limits performance due to LiDAR sparseness and limited training data, or they are model-free, making the resulting meshes vulnerable to artifacts and limited detail. Therefore, this work introduces SMPLify-3D, an optimization-based HMR framework that combines the richness of image data and the depth information within sparse LiDAR data to improve the 3D mesh inaccuracies of image-based HMR frameworks. The proposed framework consists of three main steps: 1) a body part visibility filter based on the 2D detected keypoints, 2) rough alignment between the mesh and the observed LiDAR point cloud using the ICP algorithm, and 3) an optimization scheme, inspired by [7], that modifies the actual pose and shape of the mesh to improve both the 3D and image alignment. SMPLify-3D is versatile compared to other methods and outperforms image-based and SMPL-compatible LiDAR-based HMR frameworks by improving the Per-Vertex-Error (PVE) with 45% and 26% on the 3DPW [66] and HumanM3 [17] datasets respectively. Multiple quantitative experiments are conducted to show the effects of LiDAR noise and sparsity on the framework's performance. Additionally, qualitative results illustrate how the proposed method achieves superior results on out-of-sample data recorded by a mobile robot. The source code for this work is available at: *https://github.com/guidodumont/SMPLify-3D*.

Contents

1 Introduction

1.1 Background

Many envision a future formed by new and advanced technologies such as artificial intelligence (AI), self-driving cars, virtual/augmented reality, and interactive robotics [46, 50, 9, 24]. However, looking at the current state of technology and its limitations, substantial advancements across various technical domains are required to achieve this vision. One of the limiting factors is the perception and understanding of the human body (pose and shape) during situations where technology and humans interact with each other, especially in robotics, autonomous driving, and virtual/augmented reality [13, 52].

To address this, the literature describes several methods for human perception, one of which is Human Mesh Recovery (HMR). HMR focuses on predicting a comprehensive 3D mesh of the observed human based on input sensor measurements. Typically, HMR frameworks predict a set of pose and shape parameters, which are then given to a parametric human body model, such as SMPL [44] or SMPL-X [54], to create a detailed human mesh with an underlying 3D skeleton. The richness of the information encoded in such a mesh, including 3D pose, occupancy, and optionally facial expression and hand pose, distinguishes HMR from other human perception techniques.

1.2 Problem statement

The majority of HMR frameworks only use image data to reason about the corresponding pose and shape parameters [30, 41, 8]. Despite the richness of this data, image-based HMR frameworks are vulnerable to depth ambiguity, resulting in mesh inaccuracies; mainly within the 3D pose and spatial placement of the mesh [41, 73]. Several works attempt to eliminate these ambiguities using dense 3D scans or RGBD data. However, the quantity of depth information required by such frameworks is not achievable in most applications. Only a few frameworks [38, 67, 18] use sparse LiDAR data, which is representative for robotics applications, in an attempt to improve the 3D accuracy of the mesh. However, these frameworks are either only LiDAR-based, which limits performance due to LiDAR sparseness and limited training data, or they are model-free, making the resulting meshes vulnerable to artifacts and limited detail. Therefore, we introduce an HMR framework that aims to combine both the richness of image data and the depth information within sparse LiDAR data to improve the depth ambiguity within SMPL predictions made by image-based HMR frameworks, see Figure 1.

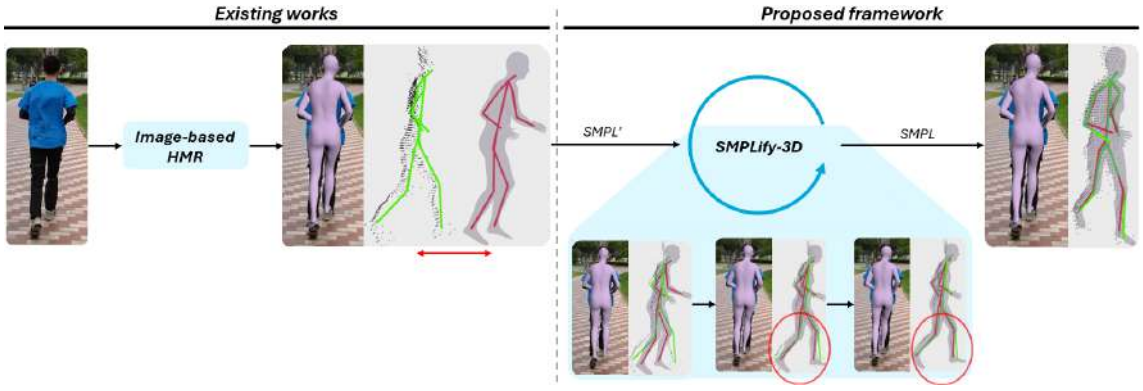


Figure 1: Image-based Human Mesh Recovery (HMR) frameworks predict a 3D human mesh based on a single image. This mesh is often *only* aligned in image space but is inaccurate in 3D space due to depth ambiguity. The proposed HMR framework, called SMPLify-3D, combines image features and sparse LiDAR data to modify the 3D pose, shape, and placement of this mesh and enhance both the image and 3D alignment. The red arrow and ellipses indicate the misplacement and 3D pose inaccuracies, respectively. Additionally, the green and red skeletons represent the ground truth and predicted 3D pose of the mesh.

1.3 Contribution

The proposed HMR framework, called SMPLify-3D, consists of three main steps and extends the optimization scheme proposed by [7]. First, a visibility filter determines which body part(s) of the human body are visible within the camera and LiDAR measurements. Next, the spatial placement of the mesh is improved by aligning the visible vertices of the mesh with the observed LiDAR points using the Iterative Closest Point (ICP) algorithm. Finally, an optimization scheme inspired by [7] optimizes the underlying pose and shape parameters of the SMPL mesh to enhance both the 3D and image alignment. Compared to [7] this optimization scheme includes two additional terms in the objective function: a term to improve the 3D alignment between the mesh and the observed LiDAR point cloud and a pose-prior to constrain the movement of occluded body parts.

The proposed HMR framework is evaluated on three popular HMR datasets and subjected to an ablation study and multiple experiments to show the effects of LiDAR noise and sparsity on the framework's performance. Additionally, qualitative results illustrate how the proposed framework achieves superior results on out-of-sample data recorded by a mobile robot, highlighting its versatility.

This work makes the following **contributions**: it presents a novel human mesh recovery (HMR) framework that leverages both image and LiDAR data to predict SMPL-compatible human meshes. By combining these data sources, the framework improves the mesh placement and 3D pose inaccuracies commonly found in image-based HMR frameworks. Additionally, we demonstrate the feasibility of deploying the proposed framework on a real mobile robot.

1.4 Document structure

This thesis is organized into several chapters to provide a comprehensive understanding of the research conducted. The related work, Chapter 2, discusses existing methods relevant to ours, highlighting their strengths and limitations. Chapter 3 describes the body part visibility filter, ICP alignment, and optimization scheme used within the proposed HMR framework. After which, Chapters 4 and 5 outline the experiments conducted to evaluate the performance of the proposed framework using datasets with simulated and real LiDAR data respectively. Each chapter concludes with a discussion to provide further insights and context. Finally, the main findings and future works are described in Chapter 6.

2 Related work

This chapter describes multiple existing research landscapes related to our work: parametric human body models, human mesh recovery (HMR) frameworks, and HMR datasets. Therefore, this chapter is divided into three sub-sections, one for each related field.

2.1 Parametric human body models

Underlying most HMR frameworks is a parametric human body model, such as SMPL [44], to deterministically generate a detailed human mesh given a set of pose and shape parameters. The SMPL (Skinnend Muti-Person Linear) model can create a wide variety of human shapes with different sizes, and poses, using a mesh of $M = 6890$ vertices. These meshes contain 24 body joints and can be created with 72 pose parameters (24×3), 10 shape parameters, and a camera translation, denoted by $\bar{\theta}$, $\bar{\beta}$, and $\bar{t}$ respectively. These pose parameters define the underlying hierarchical skeleton of the human mesh where each joint rotation is represented by an axis-angle representation. This definition of human pose in combination with the 10 shape parameters, which model the variety of body shapes and sizes, form the input space of the SMPL model. Compared to previous parametric human body models, such as SCAPE [3], SMPL significantly improved model variety and realism.

Most HMR frameworks use an underlying parametric human body to decouple mesh creation and input space reasoning; predicting 85 SMPL parameters from a given input space is less demanding than predicting a full 3D human mesh with 6890 vertices. Furthermore, a standard mesh format enables quantitative mesh reconstruction comparison and a general annotation format for HMR datasets. However, SMPL has some limitations; the non-existing details in hands and face [54], the number of parameters [51], the naked meshes without clothing or hair [2], and the long dependency chains of the pose-corrective blend-shapes [51]. These drawbacks are improved upon in future works. The SMPL-X model [8] combines the FLAME [40] and MANO [57] models with SMPL to create fully articulated hands and expressive faces, as shown in Figure 2. The STAR model [51] decreased the number of parameters by 80% and reformulated the definition of the pose-corrective blend shapes to make deformation caused by changes in pose more local, see Figure 3. Finally SMPL+D model [2] can offset SMPL vertices to model clothing and hair.

Figure 2: Mesh created by SMPL-X with fully articulated hands and fascial expression [54].

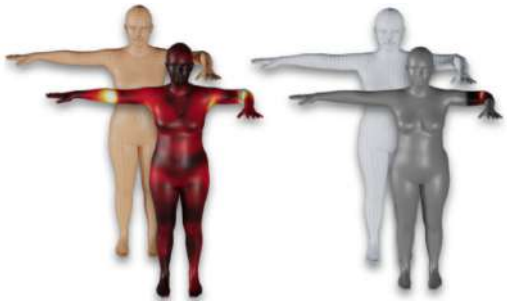


Figure 3: Corrective offset magnitudes of rotating only the left elbow in the SMPL (left) and STAR (right) model [51].

2.2 Human mesh recovery frameworks

2.2.1 Image-based HMR frameworks

Soon after the introduction of SMPL [44], Bogo [7] introduced an optimization scheme, called SMPLify, that could fit SMPL parameters to an image. The goal of the underlying objective function was to align the projected 3D joints of the SMPL model with the 2D keypoints detected in the image while keeping the over pose and shape of the mesh feasible using multiple priors. This optimization scheme was later modified by [34] to improve the mesh-image alignment of other, often deep learning-based HMR frameworks. In contrast, [29] proposed a top-down, end-to-end regression framework, that could predict SMPL parameters directly from images. To encourage feasible poses and shapes over infeasible ones, the framework consisted of a discriminator trained to recognize infeasible human bodies. This discriminator significantly contributed to the performance and was therefore copied by other end-to-end HMR frameworks such as VIBE [30]. VIBE is a popular HMR framework that can regress, temporal consistent SMPL parameters using Gated Recurrent Units (GRU) [11] based on video. All these HMR frameworks follow a top-down approach, which requires prior pedestrian detection and processes each detected human individually. In contrast, MubyNet [70] uses a bottom-up approach, which can directly regress SMPL parameters for all humans in the image without needing prior detection.

However, all image-based HMR frameworks suffer from the same depth ambiguity, due to the unconstraint transformation from 2D to 3D. This depth ambiguity makes ill-poses and global placement challenging [73]. To improve global mesh placement and rotation, [41] reformulated the projection loss and provided additional bounding box information to its proposed end-to-end regression framework called CLIFF. Although is additional information was beneficial it did not solve the global placement and rotation errors entirely. Even the first foundation model, called SMPLer-X [8], which predicts SMPL-X parameters based on images and consists of a huge backbone, trained on 4.5M instances, suffers from depth ambiguity.

In an effort to improve upon the depth ambiguity, two recently proposed HMR frameworks use the temporal information within videos to estimate camera and human motion in world frame [64, 33]. These frameworks use temporal features and optical flow to estimate camera and human motion directly using regression or within an optimization framework. However, based on the evaluation results in their publications, both models still make significant errors within mesh placement [64, 33].

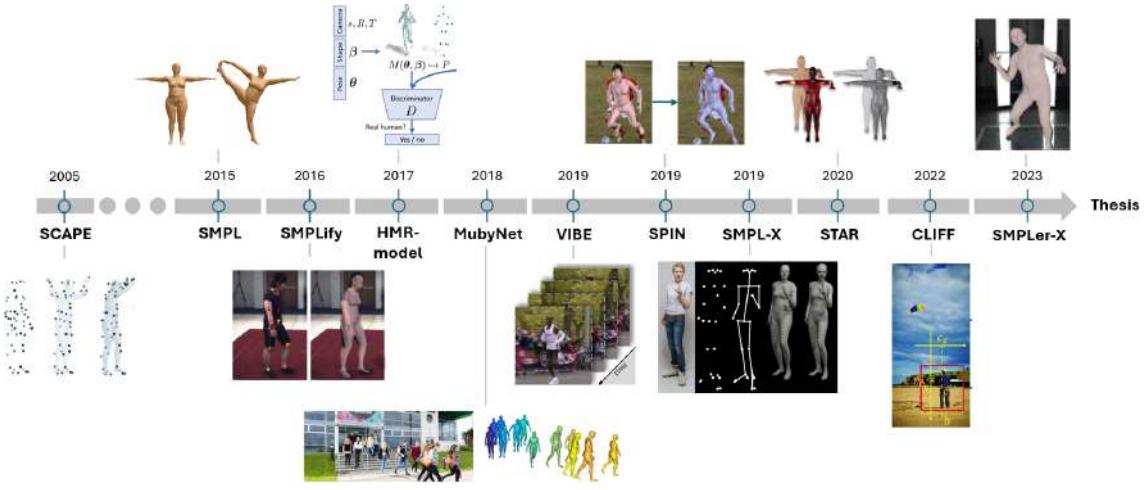


Figure 4: Evolution of image-based HMR frameworks[1]

[1]Figure 4 consists of 11 sub-figures, one for each model on the timeline. These sub-figures originate from their original papers [3, 44, 7, 29, 70, 30, 34, 54, 51, 41, 8] respectively

2.2.2 HMR frameworks based on 3D information

As described before, image-based HMR frameworks are unable to accurately reason about 3D pose and mesh placement due to depth ambiguity. However, HMR is popular within the field of computer vision and applications outside this space are yet to be discovered. Therefore, the number of publications that use (additional) 3D data to solve depth ambiguity within HMR is limited. The proposed frameworks that do, use the (additional) 3D information to 1) model detailed external surfaces based on dense 3D scans/RGBD data [58, 35] or 2) use LiDAR to improve the quality and performance of HMR frameworks in general and long-range situations [38, 67, 18]. The first research direction is mainly applicable and relatively popular in the field of computer graphics, however, research in the second direction is very limited.

LiDAR data is sparse and colorless compared to RGB(D) data, but provides valuable depth information that can improve predictions made in 3D. LiDARCap [38] and LiveHPS [56], are sequential, purely LiDAR-based, HMR frameworks that make use of this depth information by predicting SMPL-compatible human meshes in long-range scenarios. [38] showed that their model significantly outperforms image-based HMR frameworks in long-range scenarios ($>=$ 14 meters). Furthermore, they argued, based on qualitative results on the KITTI [20] and Waymo [62] datasets, that their model can make reasonable guesses on placement and pose in ambiguous situations. However, due to dataset constraints, LiDARCap only predicts pose parameters $\vec{\theta}$, meaning that the shape parameters $\vec{\beta}$ of resulting SMPL models are all constant [38]. Furthermore, although mid- to close-range performance was not provided, LiDARCap probably underperformance compared to image-based HMR frameworks in these scenarios, based on the performance vs. distance trend shown in the publication.

Similar to LiDARCap, [18] recently proposed a purely LiDAR-based model-free framework called HMR-LiDAR. This model is trained on the SLOPER4D dataset [15], among others. The pipeline consists of a 3D pose regression network followed by a mesh reconstruction network (MRN). This mesh reconstruction network increases the mesh resolution from 45 to 6890 vertices guided by the observed LiDAR point cloud [18], see Figure 5. However, the model-free nature of the resulting meshes is vulnerable to artifacts and inconsistencies, making the performance of the framework heavily dependent on both the quality and quantity of the LiDAR data. Furthermore, due to the limited amount of training data, the generalizability is limited.
To the best of my knowledge, Yang [67] proposed the only HMR framework, called S^3, which utilizes both image and LiDAR data to obtain good and consistent performance. The S^3 framework can predict skeleton, pose, and shape given a single image and/or LiDAR point cloud. This model represents a 3D human mesh as a multi-dimensional neural field to generate animatable 3D humans with clothing and texture. Based on ablation studies and qualitative results, [67] shows that combining image and LiDAR data enhances performance, especially in occluded and ambiguous scenarios. In general, LiDAR data captures global structures with depth information while images provide additional data to refine pose and shape details [67]. However, S^3 is trained on the RenderPeople dataset [1], which contains highly detailed 3D human models and images captured in a studio environment with uniform backgrounds and perfect lighting conditions. Consequently, S^3 does not apply to our research scope since its training data does not contain real-world, in-the-wild scenarios. Furthermore, the framework and pre-trained models are not open-source.

Within this thesis, we propose an optimization-based HMR framework called SMPLify-3D, which combines the richness of image data and the depth information within sparse LiDAR data to eliminate the depth ambiguity within image-based HMR frameworks.

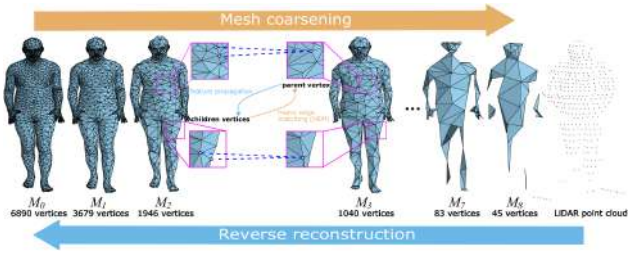


Figure 5: Principle of reverse mesh reconstruction used in the mesh reconstruction network of LiDAR-HMR [18]

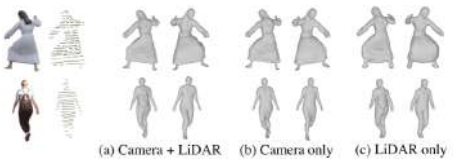


Figure 6: Qualitative results of S3 framework with different sensory inputs [67]

2.3 Datasets and benchmarks for HMR

The datasets and benchmarks used in HMR can be divided into three categories: in-the-wild (ITW), synthetic, and studio/indoor motion capture setups. Ideally, HMR frameworks are trained and tested on in-the-wild datasets with ground truth SMPL or SMPL-X annotations. However, such annotations are difficult to obtain making the datasets in this category relatively small [29, 30]. Therefore, many state-of-the-art HMR frameworks also include synthetic or 2D human pose datasets within there training data [8]. Additionally, studio/indoor motion capture datasets can be used to further diversify the training data, see Figure 7. By combining these different techniques/datasets image-based HMR frameworks can achieve robust and generalizable performance in out-of-sample scenarios [8].

In contrast, the number of datasets suitable for LiDAR-based HMR frameworks is very limited due to the computer vision-focused research environment. Consequently, the training datasets for LiDAR-based HMR frameworks lack variety in human pose, environments, distance to subject, LiDAR sensors, LiDAR sparseness, and viewpoints, see Figure 19. Moreover, the quantity and overall structure of LiDAR data can differ significantly between different instances and sensors. This makes it challenging for LiDAR-based HMR frameworks to obtain a training dataset that is large and diverse enough to achieve robust and generalizable performance. This limitation is demonstrated in the qualitative results of Figures 24 and 25 later in the thesis.

The most popular datasets for HMR are listed in Table 1

	Name	Category	Annotations		Multi-person	LiDAR-data	# images
			SMPL	*SMPL-X*			
1.	3DPW [66]	ITW	Yes	NeA	✓	✗	22.7K
2.	Talkshow [69]	ITW	-	Yes	✗	✗	3.3M
3.	UP3D [37]	ITW	Psuedo	-	✗	✗	7.1K
4.	OCHuman [75]	ITW	EFT	-	✓	✗	2.5K
5.	UBody [42]	ITW	-	Yes	✗	✗	1M
6.	SLOPER4D [15]	ITW	Psuedo*	-	✗	✓	100K+
7.	AGORA [53]	Syn	Yes	Yes	✓	✗	106.7
8.	BEDLAM [6]	Syn	-	Yes	✓	✗	951K
9.	SPEC [32]	Syn	Yes	-	✗	✗	72K
10.	SynBody [68]	Syn	-	Yes	✓	✗	633K
11.	FIT3D [19]	Studio	-	Yes	✗	✗	1.7M
12.	Human3.6M [25]	Studio	Yes	NeA	✗	✗	312K
13.	EgoBody [74]	Studio	Yes	Yes	✓	✗	845K
14.	EHF [54]	Studio	-	Yes	✗	✗	100

*The SMPL annotations only contain pose parameters

Table 1: Overview of popular HMR datasets, annotation types: EFT [28], and NeA: NeuralAnnot [48]

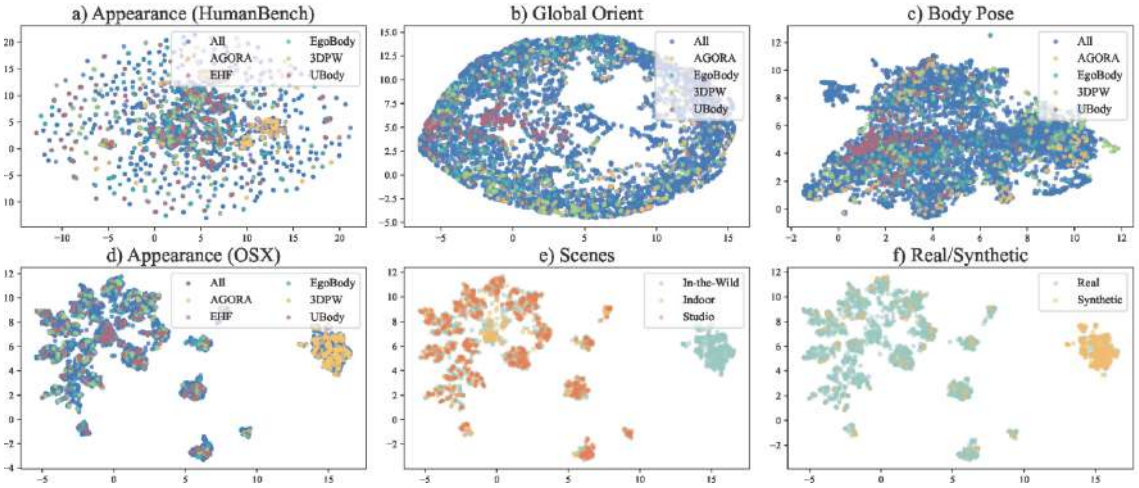


Figure 7: Dataset attribute distribution for image-based HMR-frameworks: a) and d) are image features extracted by HumanBench [65] and OSX [42] pre-trained ViT-L backbone. b) Global orientation (represented by rotation matrix) distribution. c) Body pose (represented by 3D skeleton joints) distribution. Both e) scenes and f) Real/Synthetic are drawn on the same distribution as d). All: all datasets. UMAP [45] dimension reduction is used with the x and y-axis as the dimensions of the embedded space (no unit). [8]

3 Methods

The proposed HMR framework in this work, called SMPLify-3D, aims to combine both the richness of image data and the depth information within sparse LiDAR data to improve SMPL predictions made by image-based HMR frameworks. This framework is partially inspired by [34], who used a modified version of SMPLify [7] to *only* improve the image-mesh alignment of SMPL predictions. In contrast, SMPLify-3D enhances both image and 3D alignment using image features and sparse LiDAR data.

The proposed framework consists of three main steps: 1) defining the visibility of all body parts, 2) initial LiDAR-mesh alignment, and 3) mesh optimization, see Figure 8. Each step is explained in detail in its respective section within this chapter (see Sections 4, 3.3, and 3.4). Additionally, Section 3.1 describes the image-based HMR model, called CLIFF [41], which is used for the initial SMPL predictions within this thesis.

3.1 Initial SMPL prediction

SMPLify-3D aims to enhance SMPL predictions made by image-based HMR frameworks. The input space of our proposed framework consists of this initial SMPL prediction, the observed LiDAR point cloud, and the 2D keypoints detected in the image. Theoretically, all SMPL-compatible HMR frameworks described in the literature can provide the initial SMPL prediction. However, competitive top-down regression frameworks are preferred to ensure a strong initial guess for the mesh optimization and make the overall framework more versatile by keeping it single-instance-based. CLIFF [41] aligns with all these preferences and is therefore the model chosen to provide the initial SMPL predictions within this thesis.

The Carry Location Information in Full Frames (CLIFF) model proposed by [41], aims to improve placement and pose predictions. Most top-down HMR frameworks only use an image patch as input. However, this cropped patch discards location and rotation information with respect to the camera coordinate system, limiting the accuracy of both mesh placement and 3D pose predictions [41]. To address this problem, CLIFF [41] modified the model structure of [29] by 1) extending the input space with additional bounding box information and 2) calculating the 2D reprojection loss relative to the full image rather than the cropped image patch. These modifications enabled CLIFF to achieve state-of-the-art performance at the time of publication.

Using the current state-of-the-art model PLIKS [60], was considered. However, this framework is partially optimization-based which would further increase computation time during inference.

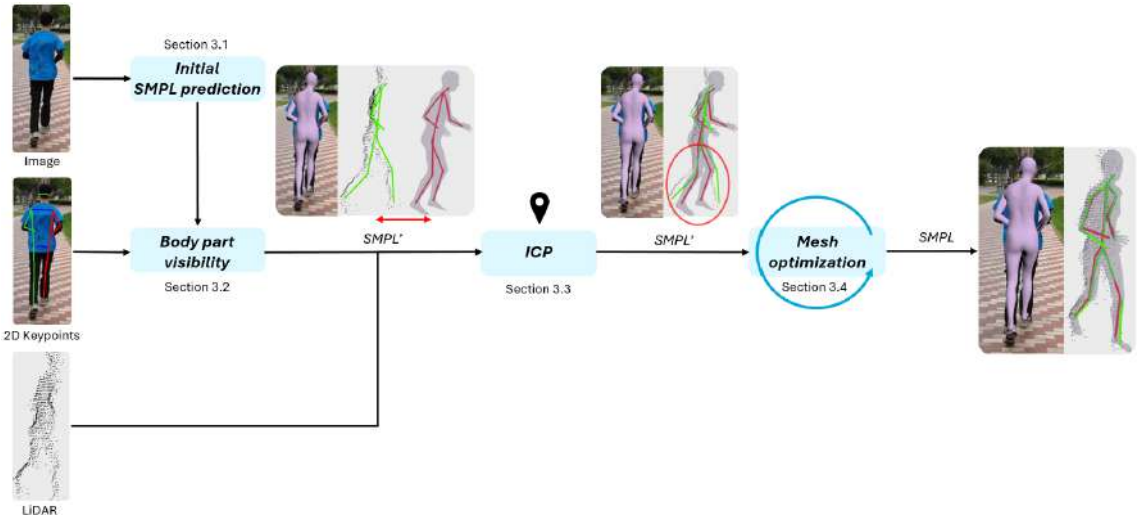


Figure 8: The proposed SMPLify-3D framework that combines the richness of image data and the depth information within sparse LiDAR data to improve SMPL predictions made by purely image-based HMR frameworks, see the red arrow and ellipse. Furthermore, the final SMPL mesh aligns in both image and 3D space.

3.2 Body part visibility

During inference, many instances contain fully or partially occluded body parts [66, 31], either due to obstacles or the camera's viewing frustum. These occlusions are a limiting factor for most HMR frameworks, but only several publications aim to improve in such situations [49, 55].

In this work, defining body part visibility is crucial for both LiDAR-mesh alignment and mesh optimization. For instance, when a pedestrian walks by, they can be rotated by 90 degrees relative to the vertical axis of the camera frame. This rotation causes one side of the body to be (partially) occluded and unobserved in both camera and LiDAR data. In contrast, the initial SMPL prediction made by CLIFF does reason about this side of the body. Therefore, directly comparing all vertices of the predicted mesh with the LiDAR data results in sub-optimal or infeasible optimization solutions in 3D space.

The backface culling algorithm [72], used in computer graphics to reduce rendering times of large-scale 3D meshes, partially addresses this problem. However, this algorithm defines face visibility purely based on the magnitude of the inner product between the face-normal $\vec{n}$ and the normalized viewing vector $\vec{p}$;

$$\vec{n} \cdot \vec{p} < w \begin{cases} \text{visible}, & \text{if true} \\ \text{invisible}, & \text{otherwise} \end{cases} \tag{1}$$

Where w is a positive threshold between 0 and 1, that defines how much a face-normal should align with the viewing vector to be classified as visible [72]. While effective, backface culling only considers face orientation and does not account for occlusions due to the object's shape itself (self-occlusion) or other objects in the scene (scene-occlusion). Variations of this algorithm exist that partially address occlusions based on the viewing frustum and other meshes in the scene. However, these variations are primarily designed for large-scale 3D scenes and focus only on scene-occlusion rather than self-occlusion [23]. Therefore, we propose a body part visibility filter specifically for SMPL meshes that removes faces occluded by self-occlusion.

The proposed filter uses the confidence or visibility scores of 2D human pose predictions to reason about body part visibility. For instance, if both the right shoulder and elbow have confidence/visibility scores below a given threshold J_{conf}, the filter assumes that the right upper arm is occluded. If a body part is identified as occluded, all faces belonging to that body part are also assumed to be occluded and thus removed from the backface culling prediction, see Figure 9. Since our work is compatible with SMPL, the correspondence between face-id and body part is known in prior and independent of pose and shape.

Although simple, this body part visibility filter improves performance for instances with (partially) occluded body parts, as demonstrated in Section 4.4. Other options, such as depth analysis, were considered. However, using the initial SMPL predictions to determine face visibility assumes that the corresponding 3D pose is partially correct, which is not always the case.

3.3 Initial LiDAR-mesh alignment

As described in the related work (see Chapter 2), image-based HMR frameworks find it difficult to accurately place the predicted mesh within the camera frame. To address this limitation, the second step of our proposed frameworks finds an initial transformation $T(R(\vec{\theta}_{glob})|\vec{t})$ between the visible vertices of the mesh $V \subset M$ and the sparse LiDAR data P using the Iterative Closest Point (ICP) algorithm [5]. After applying this transformation, the mesh is correctly placed in 3D space and can be projected onto the image using the actual camera parameters rather than a predicted set. However, the applied transformation is often so significant that the mesh scaling becomes infeasible after image projection. The mesh optimization scheme described in the next section will improve this infeasible scaling and 3D pose to improve both image and 3D alignment.

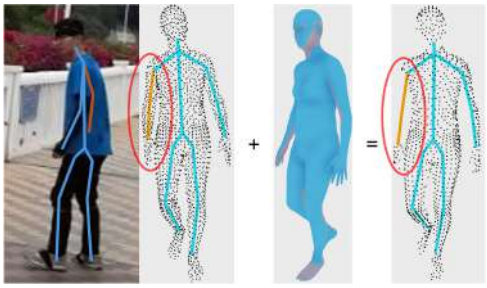


Figure 9: The visibility filter determines body part visibility based on 2D keypoint confidence scores. In this example, the right arm (indicated in orange) is marked as occluded due to low confidence. Combined with the back face culling algorithm (blue vertices in the middle image), only the visible points, as shown in the right image remain.

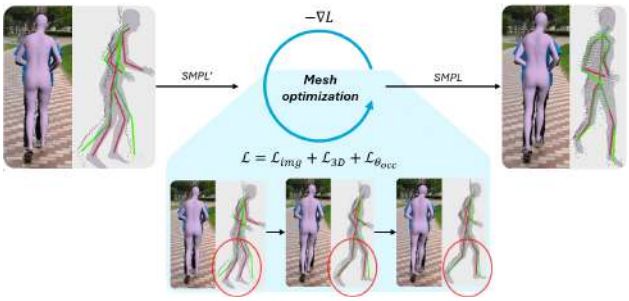


Figure 10: The mesh optimization step within SMPLify-3D aims to improve both the image and 3D alignment of the mesh by modifying the SMPL parameters, see the red ellipses.

3.4 Mesh optimization

The ICP algorithm described in the previous section aligns the predicted mesh with the corresponding LiDAR data given the initial SMPL parameters. However, the 3D pose of the initial SMPL parameters still contains inaccuracies due to depth ambiguity and ill-poses [73], see Figure 10. Therefore, both the image and 3D alignment can be improved further by optimizing all SMPL parameters; shape $\vec{\beta}$, body pose $\vec{\theta}_{body}$, camera translation $\vec{t}_{cam}$, and rotation $\vec{\theta}_{glob}$ over the observed 2D keypoints and the sparse LiDAR data respectively. This section is divided into two sub-sections defining the objective function used and the scalar weights optimization.

3.4.1 Objective function

The proposed objective function is inspired by the work of [7, 34] and introduces two additional terms 1) a term to improve the 3D alignment between the mesh and the sparse LiDAR data $\mathcal{L}_{3D}$ and 2) a pose prior that constrains the movement of occluded body parts $\mathcal{L}_{\theta_{occ}}$. The full objective function used within SMPLify-3D is defined as $\mathcal{L}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}, P, V) =$

$$\underbrace{\mathcal{L}_J(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}) + \lambda_\theta \mathcal{L}_\theta(\vec{\theta}) + \lambda_a \mathcal{L}_a(\vec{\theta}) + \lambda_\beta \mathcal{L}_\beta(\vec{\beta})}_{\text{Proposed by [7]}} + \underbrace{\lambda_{3D} \mathcal{L}_{3D}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, P, V) + \lambda_{\theta_{occ}} \mathcal{L}_{\theta_{occ}}(\vec{\theta})}_{\text{Proposed terms}} \tag{2}$$

With camera translation $\vec{t}_{cam}$, camera parameters K, the observed 2D keypoints J_{est}, the set of pose parameters $\vec{\theta} = \{\vec{\theta}_{global}, \vec{\theta}_{body}\}$, the sparse LiDAR point cloud P, visible vertices $V \subset M$, and the scalar weights λ_i.

Originally, [7] also included an interpenetration term within the objective function. However, future research [34] showed that this term does not contribute significantly to overall performance while being relatively computationally expensive. Consequently, many subsequent works, including ours, discard this interpenetration term.

Joint reprojection loss $\mathcal{L}_J$ [7]:
The objective of this term is to improve the mesh-image alignment by penalizing the weighted 2D distance between the observed 2D keypoints J_{est} and the projected 3D joints of the SMPL mesh [7]:

$$\mathcal{L}_J(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}) = \sum_{joint\ i} w_i \rho(\Pi_K(R_{\vec{\theta}}(J(\vec{\beta})_i), \vec{t}_{cam}) - J_{est,i}) \tag{3}$$

With the projection from 3D to 2D Π_K, induced by the camera parameters K, and the contribution weights for each joint w_i equal to the confidences provided by the 2D Human Pose Estimation (HPE) model. Occluded joints generally have lower confidence scores and thus a smaller contribution within this term, making them more pose prior-driven. Furthermore, the noise within the detections is handled by a Geman-McClure penalty function ρ [21].

Pose prior loss $\mathcal{L}_\theta$ [7]:
The first pose prior $\mathcal{L}_\theta$ is trained on a large-scale CMU dataset and favors probable poses over improbable ones. This prior is represented as a mixture of $N = 8$ Gaussians which all produce a weight g_i used to penalize specific infeasible pose parameters, see [7] for further definition.

$$\mathcal{L}_\theta(\vec{\theta}) = -\log \sum_j (g_i \mathcal{N}(\vec{\theta}, \mu_{\theta,j}, \sum_{\theta,j})) \tag{4}$$

Unnatural joint bending loss $\mathcal{L}_a$ [7]:
Both the joint reprojection and 3D LiDAR alignment loss can converge to a solution that requires unnatural; hyperextending or positive bending of the elbows and knees. This term aims to prevent such behavior by penalizing infeasible pose parameters [7].

$$\mathcal{L}_a(\vec{\theta}) = \sum_i exp(\theta_i) \tag{5}$$

Shape prior loss $\mathcal{L}_\beta$ [7]:
The shape prior $E_\beta(\vec{\beta})$ within the objective function penalizes infeasible shape parameters by calculating the difference between the predicted shape parameters and the shape space of the SMPL training set. This shape space is represented by a diagonal matrix $\sum_\beta^{-1}$ containing the squared singular values of all shape parameters present in the SMPL training set [7].

$$\mathcal{L}_\beta(\vec{\beta}) = \vec{\beta}^T \sum_\beta^{-1} \vec{\beta} \tag{6}$$

3D alignment loss $\mathcal{L}_{3D}$:
The 3D pose of the initial SMPL prediction can be inaccurate due to depth ambiguity and ill-poses [73]. To improve these 3D inaccuracies, the proposed 3D alignment loss $\mathcal{L}_{3D}$ aims to align the visible vertices $V \subset M$ of the mesh with the observed sparse LiDAR data P. Because the visible vertices V depend on all SMPL parameters, this term can penalize very specific pose and shape parameters to improve the alignment of individual body parts. To achieve this, this term calculates the chamfer distance [4] between the visible vertices $V \in \mathbb{R}^{nx3}$ and the sparse LiDAR point cloud $P \in \mathbb{R}^{mx3}$. To make the 3D data term independent of the number of LiDAR points, the chamfer distance is divided by the number of points within the point cloud itself:

$$\mathcal{L}_{3D}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, P, V) = \frac{1}{n} \sum_{v_i \in V}^{n} \min_{p_i \in P} \|v_i - p_i\|_2^2 + \frac{1}{m} \sum_{p_i \in P}^{m} \min_{v_i \in V} \|p_i - v_i\|_2^2 \tag{7}$$

Pose prior for occluded body parts $\mathcal{L}_{\theta_{occ}}$:
If a body part is identified as occluded, the corresponding pose parameter is only contained by the pose-prior $\mathcal{L}_\theta$ (assuming low confidence scores on the corresponding detected key points). However, this makes these pose parameters biased towards its "general" value encoded in $\mathcal{L}_\theta$. This "general" pose often differs from the observed pose, making this behavior partially undesirable. Therefore, the proposed pose prior constrains the absolute angular distance between the optimized and initial pose parameters corresponding to occluded body parts, denoted as $\vec{\theta}'_{body,occ}$ and $\vec{\theta}_{body,occ}$ respectively.

$$\mathcal{L}_{\theta_{occ}}(\theta) = \sum_{i=0} 1 - \langle q_i, q'_i \rangle^2 \tag{8}$$

where q'_i and q_i are normalized quaternion representations of the occluded pose parameters $\vec{\theta}'_{body,occ}$ and $\vec{\theta}_{body,occ}$, and $\langle \cdot \rangle$ is the inner product.

The objective function (see Equation 2) and SMPL model [44] are both fully differentiable. Consequently, the optimization problem is solved using gradient descent and PyTorch.

3.4.2 Scalar weights optimization

The optimization scheme has multiple hyperparameters; five scaler weights within the objective function (see Equation 2) $\lambda_\theta, \lambda_a, \lambda_\beta, \lambda_{3D}, \lambda_{\theta_{occ}}$, the Geman-McClure penalty ρ, backface culling threshold w, and joint confidence threshold J_{conf}. The optimal values for these hyperparameters are unique for every instance/dataset and depend on the 3D pose and shape of the humans observed and noise levels within the input data. To find the (sub)-optimal set of scaler weights for the datasets used in this work, we performed Bayesian Optimization on a subset of data.

Bayesian Optimization (BO) [59] is an efficient and robust method for hyperparameter optimization, particularly well-suited for scenarios where objective function evaluations are computationally expensive. By constructing a probabilistic surrogate model, typically using Gaussian Processes, BO predicts the objective function's behavior and iteratively selects promising hyperparameter configurations through an acquisition function that balances exploration and exploitation. This method reduces the number of evaluations needed to identify (sub)-optimal parameters, making it suited for large-scale and resource-intensive hyperparameter optimization tasks [59].

The (sub-)optimal hyperparameters found in 100 iterations of Bayesian Optimization using a subset of the data are visible in Table 2.

Dataset	ρ	λ_θ	λ_a	λ_β	λ_{3D}	$\lambda_{\theta_{occ}}$	w	J_{conf}
3DPW [66]	100	2.2	11	5	800	35	0.2	0.6
SLOPER4D [15]	100	0.75	8.5	1	500	55	0.2	0.7
HumanM3 [17]	100	1	3	17.5	600	135	-1	0.6
UTCampus[1] [71]	100	1.4	10	10	300	50	0.2	0.7

[1]UTCampus dataset does not contain SMPL annotations, the (sub)-optimal scaler weights and thresholds are determined based on qualitative results.

Table 2: (sub)-optimal scaler weights and thresholds found for the 3DPW [66], SLOPER4D [15], HumanM3 [17], and UT Campus [71] dataset

3.5 Discussion

The proposed framework, SMPLify-3D, combines image features, sparse LiDAR data, and SMPL predictions from image-based HMR frameworks to derive SMPL parameters that accurately represent observed pedestrians in both image and 3D spaces. The proposed framework is partially inspired by SMPLify [7] and consists of three main steps: 1) defining the visibility of all body parts, 2) initial LiDAR-mesh alignment, and 3) mesh optimization.

Our framework and the use of multi-modal data introduce several challenges. Firstly, the algorithm for identifying body part visibility heavily relies on the performance and confidence scores of the 2D human pose estimation (HPE) model. Consequently, inaccuracies in the detection model can lead to misidentification of occlusions, propagating errors throughout the subsequent steps. Additionally, confidence scores of 2D HPE models are not always reliable indicators of prediction accuracy and/or occlusions [27]. Secondly, the effectiveness of the Iterative Closest Point (ICP) algorithm used for initial mesh alignment is correlated with both the occluded body part detection and LiDAR noise levels. These correlations can cause the ICP algorithm to converge to suboptimal solutions. However, the subsequent mesh optimization can still mitigate potential suboptimal alignments. Thirdly, obtaining a (sub-)optimal set of hyperparameters is complex. These hyperparameters directly influence the effectiveness of the optimization scheme. The (sub-)optimal set depends on the quality of the input data and the variety of pedestrian poses observed, making it challenging to find universally effective values. Lastly, accurate optimization requires precise alignment between image features and LiDAR projections, a condition that could be difficult to consistently achieve in real-world scenarios. To address this, multiple modifications and solutions are proposed in Chapter 5 to reduce the dependency on accurate alignment between image and LiDAR data.

4 Experiments with simulated LiDAR data

This chapter presents the experiments conducted to demonstrate the performance and robustness of our proposed method, SMPLify-3D. It includes both quantitative and qualitative comparisons with other HMR frameworks, as well as experiments examining the effects of LiDAR density and noise on the framework's performance, and an ablation study. All experiments in this chapter were performed on the 3DPW dataset [66] using simulated LiDAR data and the evaluation metrics below:

- **PVE*** [*mm*] (Per Vertex Error) calculates the average distance between the predicted and ground-truth vertices.
- **MPJPE*** [*mm*] (Mean Per Joint Position Error) calculates the average distance between each predicted and ground-truth 3D joint.
- **PA-MPJPE** [*mm*] (Procrusted-Aligned Mean Per Joint Position Error) performs Procrustes alignment before computing MPJPE, which mainly measures the articulated poses, eliminating the discrepancies in scale and global rotation.

* Image-based HMR frameworks first align the predicted and ground-truth root joint to eliminate the global translation error, meaning that the shown performances of these models cannot be achieved in real life.

LiDAR data simulation
Due to the uncommon use of additional LiDAR data in HMR, none of the representative in-the-wild datasets provide synchronized image and LiDAR data with ground truth SMPL annotations [2]. Consequently, we have simulated LiDAR data on the 3DPW dataset [66] based on the provided SMPL annotations, similar to the approach in [14], see Figure 11. To examine the effects of LiDAR density on performance, we simulated three Ouster LiDARs with different resolutions, detailed in Table 3 and Figure 12.

		Resolution		Range noise [*mm*]		Angular noise [°]		Range [*m*]	
Nr.	*Sensor type*	Vert.	Hor.	mean	std	mean	std	min	max
1.	Ouster-32	32	512	±25.0	10.0	0.0	0.01	0.5	90.0
2.	Ouster-64	64	1024	±25.0	10.0	0.0	0.01	0.5	90.0
3.	Ouster-128	128	2048	±25.0	10.0	0.0	0.01	0.5	90.0

Table 3: Specifications of simulated LiDAR sensors, all have a drop-out probability of 10%

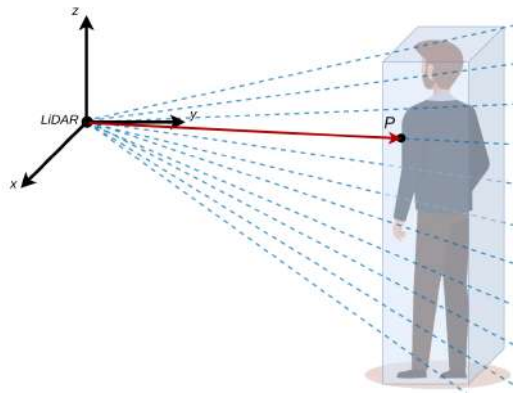


Figure 11: The calculation of the intersection point p between the LiDAR emissions (blue lines) and the mesh [14] used during LiDAR simulation.

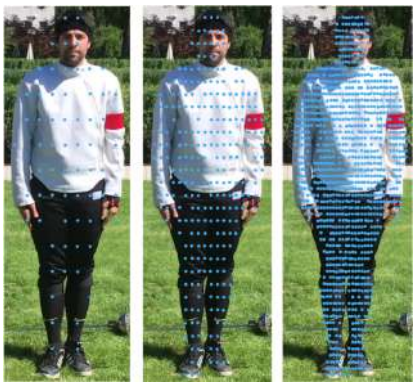


Figure 12: Simulated LiDAR with three different sensors: Ouster-32, Ouster-64, and Ouster-128 respectively

[2]The SLOPER4D dataset [15] contains synchronized image and LiDAR data, but it is not recorded in an urban environment and has limited variety in pose and distance to the root joint, see Figure 19

Besides the noise defined in Table 3, all simulated LiDAR sensors have a drop-out probability of 10% and are only defined within the camera frustum. However, due to the absence of 3D scene information, occlusions due to other entities in the scene are not accounted for within the simulation of the LiDAR data. Although this increases the available information in some situations, its effect on the final results is likely minimal due to the relatively small number of (partially) occluded instances. Furthermore, the occluded body parts algorithm (see Section) helps mitigate this issue by excluding regions of the mesh classified as occluded by the visibility filter within the mesh optimization.

4.1 Comparison results

We compared our proposed method with other, image- and video-based methods described in the literature, as shown in Table 4. All the frameworks used in this comparison did not have additional depth information. Therefore all image- and video-based models in Table 4 first align the predicted and ground-truth root joint before metric evaluation, making the presented results not achievable in real life. In contrast, the presented SMPLify-3D performance does not require root joint alignment, making them representative for real-life performance. Despite the difference in alignment, our framework outperformce image-based HMR frameworks on all metrics, independent of the LiDAR's resolution. Moreover, the improved PA-MPJPE also shows that SMPLify-3D does not only apply a translation/rotation but also modifies the pose $\vec{\theta}$ and shape $\vec{\beta}$ parameters to achieve better performance.

The performance of SMPLify-3D is correlated with the resolution of the LiDAR sensor used. Lower-resolution sensors have fewer depth measurements on the subject, resulting in weaker guidance and constraints in 3D. However, the performance gap between the Ouster-64[2] and Ouster-128[3] is significantly smaller than that between Ouster-64[2] and Ouster-32[1], suggesting a lower bound on the necessary depth information density (see further experiments in Section 4.2 and Figure 16).

To compare real life performance, we evaluated CLIFF [41] without root joint alignment and with ICP alignment on the test set of 3DPW [66], see Table 5. These results show that the performance presented in the publications are, by far, not achievable in real life and demonstrate the value of our contribution. ICP alignment between the predicted mesh and recorded LiDAR data significantly improves the raw performance of CLIFF, however, it is still underperforming compared to CLIFF with root alignment and SMPLify-3D.

		3DPW [66]		
	Method	PVE ↓	MPJPE ↓	PA-MPJPE ↓
Video	VIBE [30]	99.1	82.9	51.9
	DynaBOA [22]	82.0	65.5	40.4
	MotionBERT [76]	79.4	68.8	40.6
	WHAM-B [61]	71.0	59.4	37.2
Image	SMPLify [7]	106.8	-	-
	TRACE [64]	97.3	79.1	37.8
	SPIN [34]	96.9	59.2	-
	ROMP [63]	93.4	76.7	47.3
	HybrIK [39]	82.3	71.6	41.8
	CLIFF [41]	81.2	69.0	43.0
	Cha. [10]	76.3	66.0	39.0
	PLIKS [60]	73.3	60.5	38.5
LiDAR	SMPLify-3D [1]	73.2	57.0	47.7
	SMPLify-3D [2]	57.2	43.9	38.5
	SMPLify-3D [3]	**50.5**	**39.1**	**35.1**

Table 4: Performance comparison between SMPLify-3D with different LiDAR sensors and other image-/video-based methods on the 3DPW dataset [66].
[1 2 3] LiDAR type used following Table 3.

	Method	PW3D [66] PVE ↓	PW3D [66] MPJPE ↓
CLIFF	With root alignment	81.2	69.0
	Without root alignment	481.1	479.0
	ICP alignment[2]	103.4	90.5
LiDAR	SMPLify-3D	73.2	57.0
	SMPLify-3D	57.2	43.9
	SMPLify-3D	**50.5**	**39.1**

Table 5: Real life performance comparison between SMPLify-3D and CLIFF [41].

From the camera's perspective, the performance difference between CLIFF [41] and SMPLify-3D, as shown in Table 4, appears minimal, see Figure 13. However, the difference becomes significant when viewed parallel to the camera's depth axis. In this view, the 3D inaccuracies predicted by CLIFF are clearly visible, primarily due to ill-poses; multiple 3D poses can result in similar 2D projections [73]. With the additional depth information, SMPLify-3D can mitigate these artifacts by fitting the predicted mesh to this extra data during optimization.

Figure 13: Qualitative results of SMPLify-3D (blue) compared to CLIFF [41] (red) and the ground truth (green) on some samples of the 3DPW test-set [66]. Note that the CLIFF results are first aligned with the pelvis joint before 3D visualization.

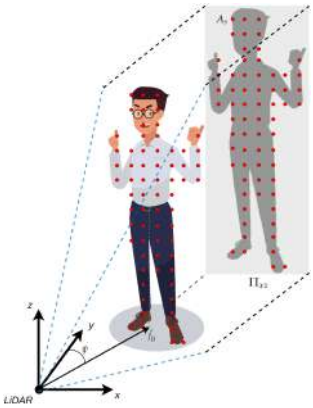


Figure 14: Area A_s of projection Π_{xz} used to calculate LiDAR density.

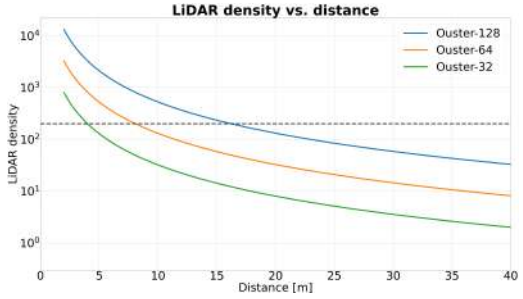


Figure 15: LiDAR density $[/m^2]$ vs distance to subject.

4.2 Effect of LiDAR density and distance to subject

The sparsity of LiDAR data, and consequently the number of LiDAR points per instance, is influenced by the sensor's resolution and the distance to the subject. Although the ICP algorithm and the 3D alignment term in the objective function (see Equation 7) are independent of the number of points, sparse depth information contains less information and is therefore correlated with performance. To see what the effect of this sparsity is, we evaluated SMPLify-3D on the 3DPW dataset [66] while recording the number of LiDAR points per instance, point cloud density $[/m^2]$, and the distance $[m]$ to subject.

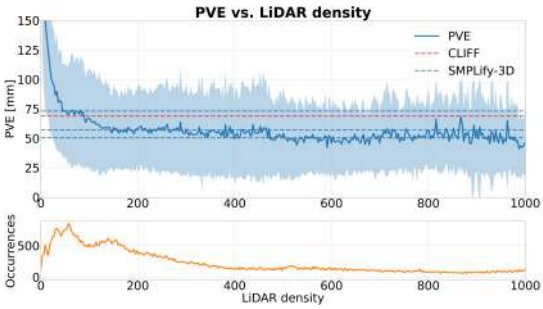
Since the number of LiDAR points per instance directly correlates with both sensor resolution and distance to the subject, different scenarios can yield the same number of points. For instance, a low-resolution sensor with the subject nearby and a high-resolution sensor with the subject farther away can have the same number of LiDAR points. To eliminate these conflicting scenarios and obtain more consistent results based solely on distance, as shown in Figure 15, we formulated point cloud density [points$/m^2$] as:

$$\rho = \frac{\text{num. points}}{A_s} \approx \frac{\text{num. points}}{A\left(\Pi_{xz}\left(R_z(\psi)M\right)\right)} \tag{9}$$

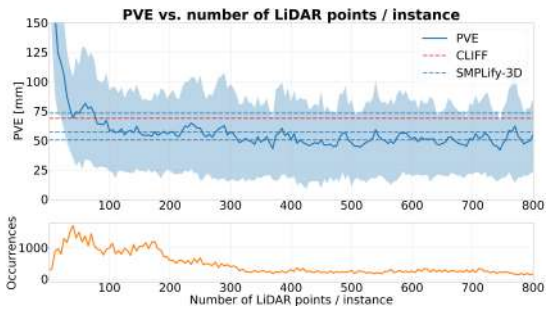
With vertices M, rotation around the z-axis $R_z(\cdot)$ with angle ψ, xz-plane projection Π_{xz}, and surface area operator $A(\cdot)$ to create the surface projection area A_s, see Figure 14.

Both PVE and MPJPE improve exponentially with increasing LiDAR density and the number of points per instance, as shown in Figure 16. This exponential trend highlights the significant value additional depth information brings to the proposed method, even in scenarios with low LiDAR density and high-instance sparsity. Moreover, despite the decreasing number of occurrences, the mean and standard deviation of both metrics converge to a constant value. This converging behavior indicates that beyond a certain threshold, additional 3D information does not further enhance performance. Based on these results, this threshold appears to be around a density of 200 points/m² or 100 points per instance. These thresholds align with observations made by [18]

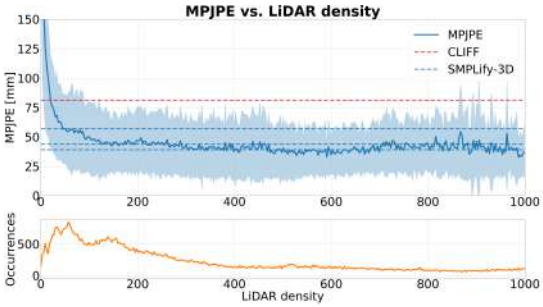
In contrast, when visualizing the metrics against the distance to the subject, as seen in Figure 17 a-b, the correlation is less pronounced. Here, performance shows a slight positive correlation with the distance to the subject, which varies depending on the LiDAR's resolution. This decline in performance and positive correlation is most evident with the Ouster-32, though it is relatively minor and noisy due to the limited number of occurrences.



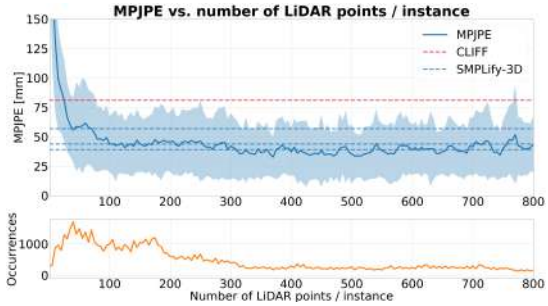
(a) PVE vs. LiDAR density [$/m^2$].



(b) PVE vs. number of points per instance.



(c) MPJPE vs. LiDAR density [$/m^2$].



(d) MPJPE vs. number of points per instance.

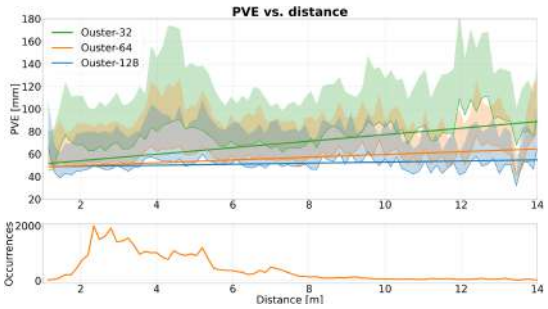
Figure 16: Results of LiDAR density experiments

The relationship between LiDAR resolution and the amount of correlation can be explained using Figure 15. The density threshold of 200 points/m^2, explained in the previous paragraph, is reached at different distances from the sensor depending on its resolution. For instance, this threshold is reached at 16.5m or 8.0m for the Ouster-128 and Ouster-64, respectively, assuming constant pose. This means that the density threshold of 200 points/m^2 is not reached for the Ouster-128 or only in a small number of instances for the Ouster-64, explaining the minimal or no correlation between performance and distance in Figure 17 a-b. Conversely, the density threshold for the Ouster-32 is reached at 4.1m. This results in a larger number of instances with insufficient LiDAR density, amplifying the correlation and making it more noticeable in Figure 17 a-b.

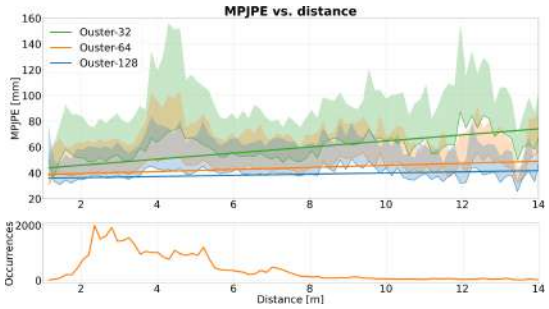
Additionally, the mean and standard deviation of both evaluation metrics are correlated with the sensor's resolution, as seen in Table 6. It is noteworthy that the magnitude of the mean and standard deviation decrease at a similar rate across the metrics. This indicates that higher-resolution sensors contribute to improved averages and reduced noise levels.

		PVE ↓		MPJPE ↓	
	Method	mean	std	mean	std
CLIFF	With ICP alignment	103.4	54.0	90.5	47.3
CLIFF	With root alignment	81.2	30.6	69.0	8.1
Ours	Ouster-32	73.2	50.9	57.0	42.7
Ours	Ouster-64	57.2	40.0	43.9	31.0
Ours	Ouster-128	50.5	35.4	39.1	26.4

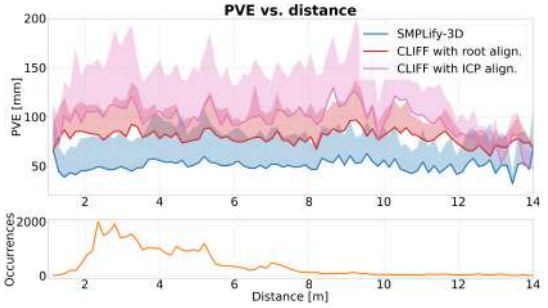
Table 6: Comparison between CLIFF [41] and SMPLIfy-3D on achieved mean and standard deviation on both PVE and MPJPE.



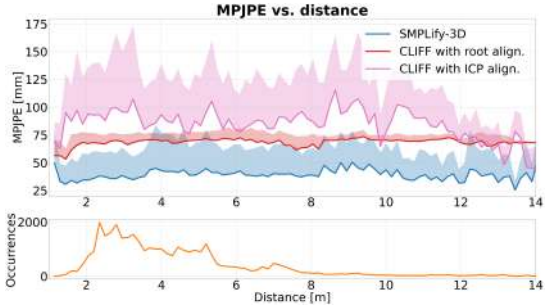
(a) PVE* vs. distance to the subject.



(b) MPJPE* vs. distance to the subject.



(c) PVE* vs. distance to subject comparison between SMPLify-3D and CLIFF with ICP and root joint alignment [41].



(d) MPJPE* vs. distance to subject comparison between SMPLify-3D and CLIFF with ICP and root joint alignment [41].

Figure 17: Correlation between performance and distance to the subject
* Due to visualization purposes, only the positive standard deviation is shown

When comparing the performance for distance to subject with CLIFF [41], we obtain Figure 17 c-d. As expected, CLIFF with root alignment shows better standard deviation (noise levels) compared to CLIFF with ICP alignment, particularly when comparing MPJPE. The small standard deviation in MPJPE for CLIFF with root alignment, seen in Figure 17 d and Table 6, originates from the evaluation metric itself. MPJPE calculates the average distance between all 3D joints, so aligning the root joint (hips) with the ground truth significantly reduces noise. In contrast, CLIFF with ICP alignment and SMPLify-3D (Ouster-128) both rely on ICP alignment, between the the visible points of the predicted mesh and the observed LiDAR point cloud. The quality of this alignment fluctuates, increasing the standard deviation. However, SMPLify-3D outperforms both versions of CLIFF in terms of average performance.

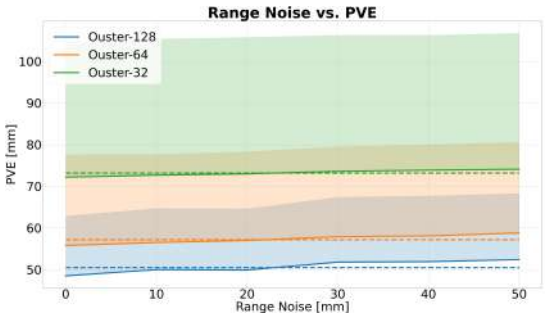
4.3 Effect of LiDAR noise

Industrial LiDAR sensors can accurately measure distance using the concept of time-of-flight [36]. However, LiDAR measurements contain noise, particularly in distance and angle measurements, the number of returned reflections, and measured intensities. [36] conducted extensive performance tests to quantify the magnitude and amount of noise for 10 commonly used LiDAR sensors, including the Ouster LiDAR's we have simulated. We used their findings to define feasible noise levels, see Table 18a, to quantify the impact of LiDAR noise on our proposed method. The noise levels are individually applied to all three simulated LiDAR sensors while keeping the other types of noise constant following Table 3. The noise is modeled using the noise definition of CARLA [16], after which the performance on the 3DPW dataset [66] is evaluated, see Figure 18.

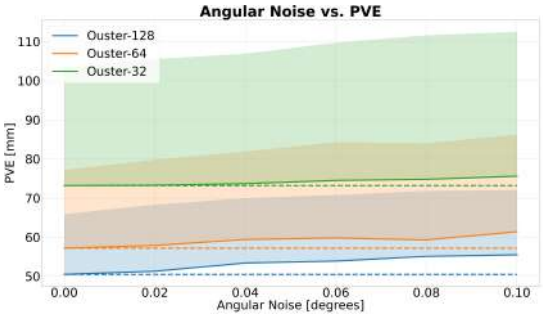
Range noise
There is a clear, almost linear, positive correlation between range noise and PVE performance, as shown in Figure 18b. The deviation between extremes slightly decreases with lower LiDAR resolution, suggesting a correlation between depth information density and the impact of range noise on performance. Sparse depth information offers less guidance and constraints, making optimization less dependent on depth. Conversely, dense depth information provides more guidance and constraints, making the optimization more sensitive to noise.

	mean			std
	min	max	step	
Range noise [mm]	0.0	50.0	10.0	10.0
Angular noise [°]	0.0	0.1	0.02	0.01
Dropout-rate	0.0	0.6	0.1	-

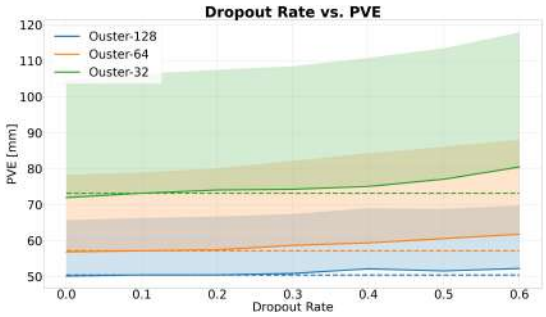
(a) Noise levels used, based on [36], within LiDAR noise experiment.



(b) Influence of range noise on the 3DPW [66] test-set.



(c) Influence of angular noise on the 3DPW [66] test-set.



(d) Influence of dropout on the 3DPW [66] testset.

Figure 18: Results of LiDAR noise experiments.

Angular noise
The overall trends observed for range noise; positive linear correlation and the connection between performance decrease and LiDAR resolution, are also visible in the results for angular noise, see Figure 18c. However, these results are noisier, possibly due to the random over- and under-estimation of the measured angle. These errors can either cancel out or amplify the overall trend, adding noise to the results.

Dropout
Compared to both range- and angular noise, the visible trend for dropout rate is significantly different, see Figure 18d. As expected, high-resolution LiDAR sensors are less affected by higher dropout rates compared to lower-resolution sensors. Furthermore, the overall positive correlation moves toward exponential, amplifying the differences between the different sensors. The correlation between LiDAR density and performance, described before, is also visible in these results, given that every LiDAR sensor improves with less/zero dropout.

4.4 Ablation study

We conducted an ablation study to investigate the significance of the newly introduced components; visible points algorithm, 3D data term E_{3D}, and the additional pose prior $E_{\theta_{occ}}$ within our proposed framework SMPLify-3D. By systematically disabling each term in isolation, we aim to understand how each component contributes to the overall performance. The ablation study was performed on the test set of the 3DPW dataset [66] and for all three LiDAR sensors defined in Table 3.

The ablation results in Table 8 show that the 3D data term E_{3D} within the objective function contributes the most to overall performance, followed by the pose prior $E_{\theta_{occ}}$ and the body part visibility filter. The constant performance without E_{3D} across all three LiDAR sensors indicates that the ICP alignment before the optimization is less sensitive to depth information density compared to the mesh optimization (see Figures 16a and 16c). A similar observation applies to the performance difference with and without the additional pose prior $E_{\theta_{occ}}$ as the pose prior is independent of depth information.

However, the contribution of the body part visibility filter is not apparent in Table 8 because the number of instances with one or more partially occluded body parts is low compared to the total number of instances in the dataset. Therefore, we repeated the ablation study for the body part visibility filter only on the instances with partially occluded body parts, see Table 7. This sampled dataset consists of 1,063 instances (2.9% of the total number of instances), each containing at least one occluded body part. Within this subset, the contribution of the body part visibility filter is more prominent, despite being less significant compared to the other terms in the ablation study.

		All terms		No body part visibility filter			
		mean	std	mean	std	t-value	p-value
Ouster -32	PVE ↓	82.0	50.9	84.3	60.2	9.512	0.3416
	MPJPE ↓	62.9	42.6	64.9	52.7	9.623	0.3360
	PA-MPJPE ↓	50.2	35.0	51.5	42.3	7.720	0.4402
Ouster -64	PVE ↓	66.9	40.0	69.3	47.6	12.585	0.2082
	MPJPE ↓	48.2	31.0	50.4	36.4	15.002	0.1337
	PA-MPJPE ↓	41.1	29.4	42.3	34.8	8.588	0.3905
Ouster -128	PVE ↓	58.7	35.4	60.2	42.2	8.879	0.3747
	MPJPE ↓	43.6	26.4	45.3	35.6	12.506	0.2112
	PA-MPJPE ↓	37.2	27.1	38.7	33.5	11.350	0.2565

Table 7: Ablation study results to understand the importance of the proposed visibility filter for the (partially) occluded body parts. This study is conducted on a sampled 3DPW dataset [66] containing 1,063 instances, all with (partially) occluded body parts.

		All terms		No $\mathcal{L}_{\theta_{occ}}$				No $\mathcal{L}_{3D}$			
		mean	std	mean	std	t-value	p-value	mean	std	t-value	p-value
Ouster -32	PVE ↓	73.2	50.9	75.9	75.9	-68.66	0.00	171.2	113.1	-191.95	0.00
	MPJPE ↓	57.0	42.6	58.2	43.2	-66.92	0.00	159.3	113.2	-194.86	0.00
	PA-MPJPE ↓	47.6	35.0	48.9	35.5	-53.33	0.00	49.6	22.1	-78.44	0.00
Ouster -64	PVE ↓	57.2	40.0	60.1	41.3	-33.2	1.09-23	173.2	117.4	-188.63	0.00
	MPJPE ↓	43.9	31.0	45.2	31.8	-27.73	0.00	161.7	117.8	-191.47	0.00
	PA-MPJPE ↓	38.5	29.4	40.4	31.0	-24.21	0.00	49.2	21.6	-77.00	0.00
Ouster -128	PVE ↓	50.5	35.4	52.4	36.8	-7.18	1.43-12	174.9	121.4	-185.41	0.00
	MPJPE ↓	39.1	26.4	40.3	27.6	-5.74	9.71-09	163.4	121.8	-188.04	0.00
	PA-MPJPE ↓	35.1	27.1	36.6	28.9	-7.27	3.62-13	49.4	23.0	-76.71	0.00

Table 8: Ablation study results to understand the importance of the proposed terms within the objective function, see Equation 2. This study is conducted on the full 3DPW dataset [66] containing 35,515 instances.

4.5 Discussion

The experiments in this chapter used simulated LiDAR data on the 3DPW dataset [66] to quantitatively compare our method with other HMR frameworks and examine the impact of LiDAR density and noise on performance. Additionally, we conducted an ablation study to highlight the contributions of the body part visibility filter and each proposed term in the objective function.

The quantitative comparison showed that our proposed method surpasses image- and video-based HMR frameworks across all metrics, independent of LiDAR resolution, see Table 4. This improvement indicates that combining the richness of image data with the depth information within LiDAR can enhance HMR performance, irrespective of LiDAR resolution. Unlike image-/video-based HMR frameworks that align the root joint with the ground-truth annotation to remove global mesh placement ambiguity before evaluation, SMPLify-3D aligns the predicted mesh with the observed LiDAR data. This makes root alignment before evaluation redundant and the presented results achievable in real life scenarios, see Table 5.

It's important to note that SMPLify-3D is one of the first HMR frameworks to combine image features with sparse LiDAR data, making the quantitative results in Table 4 not a direct one-to-one comparison. The simulated LiDAR data was created based on ground-truth human mesh annotations from the 3DPW dataset [66]. However, this data also included occluded body parts due to the absence of 3D scene information. Although the body part visibility filter filters out some of these body parts, the simulated data still is not fully representative for real-world measurements. Moreover, the simulated LiDAR points did not account for misidentified points from other entities in the scene due to occlusions, potentially affecting performance.

The effect of LiDAR density and noise on the proposed framework's performance was evaluated through multiple experiments. As anticipated, performance improved with increased LiDAR density. However, the correlation plateaued at approximately 100 LiDAR points per instance, beyond which additional points became less to not effective. Furthermore, all types of LiDAR noise negatively impacted performance, with the extent of this impact varying according to the LiDAR resolution.

Despite the extensive results, the majority of instances within the 3DPW dataset are at close range ($< 6m$), making the LiDAR experiments biased towards these close-range situations and less representative of long-range scenarios. Additionally, due to limited resources, only a few noise levels and combinations were tested, and LiDAR sensors with 16 vertical rays were not included in the experiments.

The ablation study illustrated the contribution of the visibility filter and each term within the objective function to overall performance. However, the visibility filter only contributes to performance if one or more body parts are identified as occluded. Therefore, a small subset of the dataset was sampled to specifically show the contribution of this term. Despite an improvement, the visibility filter is less significant compared to the proposed terms within the objective function. It could be more effective to use another pose estimation model that directly or more accurately reasons about joint visibility.

5 Experiments with real LiDAR data

This chapter describes the experiments conducted with real LiDAR data, in contrast to the experiments presented in Chapter 4, which utilized simulated LiDAR data. The chapter is divided into two sections. The first section provides a quantitative and qualitative comparison of our proposed method with other point cloud-based HMR frameworks using the SLOPER4D [15] and HumanM3 [17] datasets. The second section showcases the versatility of our proposed method with qualitative results on the UT Campus dataset [71] and data collected by a mobile robot in the AMR lab[3].

5.1 Comparison with other point cloud-based HMR methods

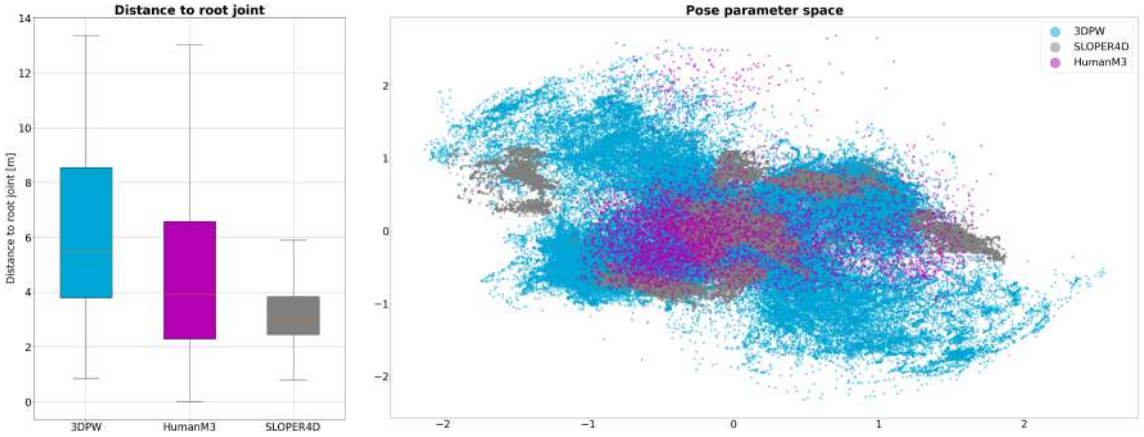
5.1.1 Datasets

We evaluated our proposed method, SMPLify-3D, against other 3D pose and HMR frameworks that use point cloud data on the SLOPER4D [15] and HumanM3 datasets.

SLOPER4D [15] is a single-person, outdoor motion capture dataset with synchronized image and LiDAR data along with ground-truth SMPL annotations. However, the dataset is only partially released and does not contain an official train- and test-split. Therefore, we follow the data splits proposed by [18], resulting in a test set with 8,064 instances sampled from a single scene.

HumanM3 [17] is a multi-view, multi-modal, and multi-person dataset for 3D pose estimation. It includes four different outdoor scenes and provides 15 3D keypoints for every person observed, resulting in a test set of 8,951 instances.

Compared to the 3DPW dataset [66] used in Chapter 4, the SLOPER4D and HumanM3 datasets exhibit different characteristics, as shown in Figure 19. The SLOPER4D test set is limited to a single sequence of a man running, viewed from behind, offering limited variety in distance and 3D poses. Conversely, the HumanM3 dataset features multiple scenes and multiple people, providing greater variety in both distance and 3D pose. Additionally, HumanM3 includes many instances where subjects are partially occluded by others, making it more challenging than the SLOPER4D dataset. However, both datasets can not match the variety in distance and 3D pose within the 3DPW dataset. This difference in variety is also visually apparent when comparing data samples, see Figure 20.



(a) Distance between the camera and root joint of the ground truth annotation.

(b) Dimensionallity reduced pose parameter $\vec{\theta}$ space of the SMPL annotations within the three datasets. Compared to SLOPER4D (gray) and HumanM3 (purple), 3DPW (blue) contains much more variety in 3D pose.

Figure 19: Data attributes of 3DPW [66], SLOPER4D [15], and HumanM3 [17] datasets.

[3] https://autonomousrobots.nl/

Figure 20: Data samples form the 3DPW [66] (top), HumanM3 [17] (middle), and SLOPER4D (bottom) [15] datasets.

5.1.2 Quantitative results

Table 9 includes both model-free and SMPL-based methods. Model-free methods predict the locations of all 6,890 mesh vertices directly, while SMPL-based methods predict a set of SMPL parameters; 72 pose and 10 shape parameters. This difference in mesh construction leads to remarks for quantitative comparison. First, model-free methods can place vertices and 3D joints anywhere in space, whereas SMPL-based methods are constrained by the SMPL model's pose and shape parameters. These parameters have long-range correlations and affect all vertices and 3D joints when a single parameter changes. This constraint prevents SMPL-based methods from making small local mesh adjustments [51]. Second, the mesh structures of model-free methods can differ significantly from parametric methods. As a result, the Per-Vertex-Error (PVE) can sometimes be misleading and may not accurately represent mesh reconstruction quality [18]. To address this, [18] introduced the Mean-Per-Edge-Relative-Error (MPERE) for model-free methods, which measures the difference between the predicted l_i and ground-truth $\hat{l}_i$ length of mesh all edges m to reason about the reconstruction quality of short edges in dense parts more effectively [18].

$$MPERE = \sum_{i=1}^{m} \frac{1}{m} \frac{||l_i - \hat{l}_i||_1}{\hat{l}_i} \tag{10}$$

		SLOPER4D [15]				HumanM3[1] [17]	
	Method ↓	PVE ↓	MPJPE ↓	PA-MPJPE ↓	MPERE ↓ [18]	MPJPE ↓	PA-MPJPE ↓
Model-free	PRN [18]	-	57.0	-	-	82.2	-
	V2V-PoseNet [47]	-	**50.7**	-	-	83.0	-
	PRN+P2M [18, 12]	65.3	56.6	-	0.132	80.6	-
	V2V+P2M [47, 12]	59.8	**50.7**	-	0.126	83.0	-
	LiDAR-HMR [18]	**51.9**	51.0	-	**0.094**	**77.6**	-
SMPL-based	LiDARCap [38]	148.1	158.3	-	**0.050**	175.8	-
	SAHSR [26]	81.2	72.6	-	0.085	105.5	-
	VoteHMR [43]	**60.9**	**54.6**	-	0.079	105.8	-
	CLIFF [41]	93.7	84.7	69.3	0.057	106.3	66.5
	SMPLify-3D	64.4	55.8	**43.5**	0.060	**83.9**	**58.1**

[1] Multi-view

Table 9: Performance comparison between SMPLify-3D other point-cloud-based 3D pose and HMR frameworks on the SLOPER4D [15] and HumanM3 [17] datasets.

Due to these differences in mesh structure, comparing reconstruction quality using PVE and MPERE can be difficult. However, comparing 3D pose accuracies with MPJPE and PA-MPJPE provides a better performance representation across both method types. Given the flexibility differences, it is unsurprising that SMPLify-3D is slightly outperformed by most model-free methods.

Furthermore, SMPLify-3D is optimization-based, meaning the presented performance is achieved by only adjusting hyperparameters, unlike other models (except CLIFF) that are specifically trained on the corresponding dataset. This demonstrates the versatility of our method making it potentially more suited for different unseen inference data.

SMPLify-3D is the only SMPL-based HMR framework that achieves competitive performance on both datasets. While VoteHMR [43] is competitive on the SLOPER4D dataset [15], its performance drops significantly on the more challenging HumanM3 dataset [17]. This trend is consistent across all SMPL-based HMR frameworks, underscoring the superiority of SMPLify-3D. Additionally, the performance gap between CLIFF [41] and SMPLify-3D highlights the value of incorporating LiDAR data within HMR and the contributions of this work.

5.1.3 Qualitative results

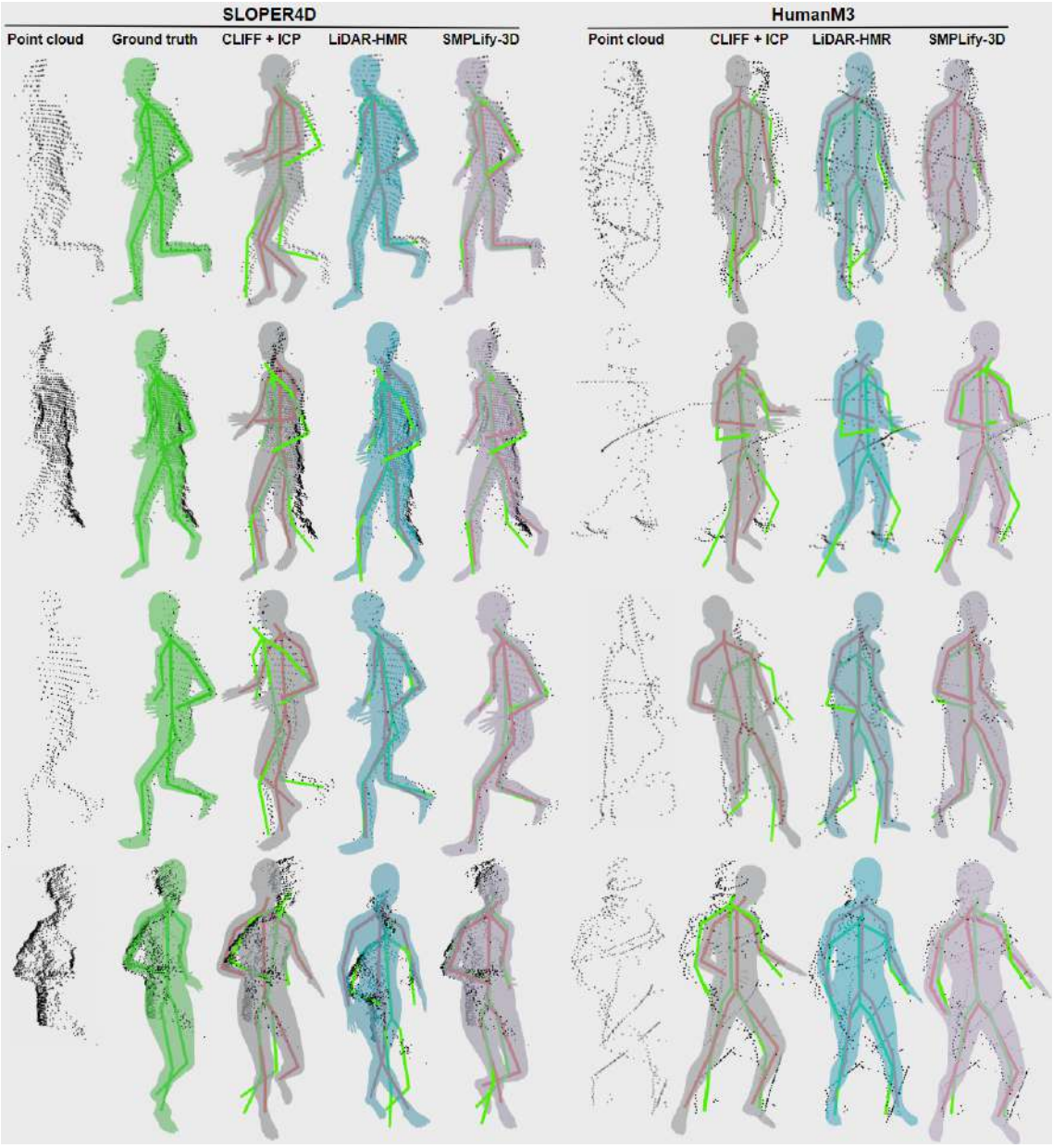


Figure 21: Qualitative comparison between CLIFF [41] with ICP alignment (gray), LiDAR-HMR [18] (blue), and SMPLify-3D (purple) on the SLOPER4D [15] and HumanM3 [17] datasets. The ground truth mesh and 3D pose annotations are shown in green and the predicted 3D poses are illustrated in red.

Figure 22: The LiDAR and front-facing camera of the Jackal robot are used in lab experiment

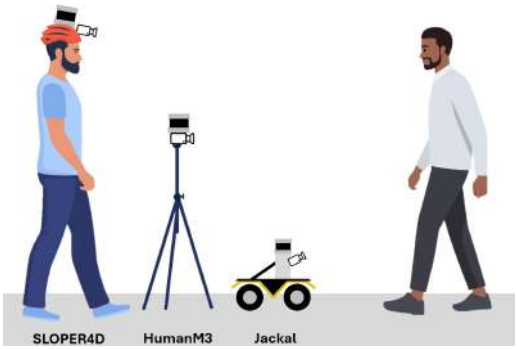


Figure 23: Difference in viewpoint across the datasets used [15, 17] and the lab experiment

The competitive performance indicated in Table 9 is also evident in the qualitative results shown in Figure 21. Generally, instances with substantial LiDAR coverage from top to bottom appear visually comparable. However, in cases where the LiDAR data is sparse or does not cover all body parts, SMPLify-3D shows superior qualitative performance. Additionally, as seen in the fourth row of Figure 21, LiDAR-HMR can predict infeasible joint angles. Moreover, the effectiveness of our optimization scheme is prominent when comparing the CLIFF + ICP results with the SMPLify-3D mesh predictions.

5.2 Qualitative experiments

To further evaluate the versatility of our proposed HMR framework, we conducted two additional qualitative experiments using data recorded by a mobile robot. The first subsection presents qualitative results from data recorded by a mobile robot named Jackal in the AMR lab. The second subsection showcases qualitative results on the UT Campus dataset and introduces a minor modification to the original optimization scheme proposed in Chapter 3. This minor modification aims to improve performance in scenarios where the LiDAR-image alignment is sub-optimal.

5.2.1 Experiment in the lab

The data for this lab experiment was recorded by a mobile robot named Jackal, see Figure 22. This robot is designed for navigating crowded urban environments and is equipped with a VLP16 LiDAR and five Intel RealSense D455 cameras arranged in a circular formation to provide 360-degree camera coverage. However, this experiment only utilized the front-facing camera.

The data recorded by the Jackal robot differs significantly from the data with the SLOPER4D [15] and HumanM3 [17] datasets for two reasons; 1) Due to the small footprint of this mobile robot, both the LiDAR sensor and camera are mounted significantly closer to the ground, as illustrated in Figure 23. 2) The VLP16 LiDAR sensor, with its 16 vertical rays, has a lower resolution compared to the sensors used in the other datasets, resulting in much sparser LiDAR data.

The dataset used within this experiment contains 5 minutes of data across two sequences involving single and multi-person scenarios, resulting in 4,182 instances. Unfortunately, due to time and complexity constraints, the recorded data does not include 3D pose or SMPL annotations.

The qualitative results presented in Figure 24 demonstrate the performance differences between LiDAR-HMR [18] and the proposed method. In straightforward scenarios where the observed human is standing upright facing the camera/LiDAR (first row), both models can predict feasible 3D pose and shape. However, in scenarios with unique or less common 3D poses, the LiDAR-HMR models often fail to predict feasible HMR results; some predictions face the wrong direction (see row 2) or exhibit infeasible 3D pose or shape (see rows 3 and 4). In contrast, SMPLify-3D accurately models such unique or uncommon 3D poses. Additionally, by utilizing image features, SMPLify-3D significantly improves image-mesh alignment compared to LiDAR-HMR and enhances robustness against noise in the LiDAR data (see the left person in row 5). These results indicate that pre-trained LiDAR-HMR models are biased towards the 3D poses and shapes present in their corresponding training set, leading to limited performance on out-of-sample data.

In contrast, SMPLify-3D is more versatile and able to produce feasible predictions even in situations with challenging 3D poses and sparse LiDAR data, highlighting its superior adaptability and robustness.

Figure 24: Qualitative comparison between two LiDAR-HMR [18] models and SMPLify-3D on data recorded within the AMR-lab. The two LiDAR-HMR models are respectively trained on the SLOPER4D [15] and Waymo [62] datasets.

5.2.2 UT Campus

Finally, the proposed method is qualitatively evaluated on the UT Campus dataset [71]. This dataset, combined with the presented method, will be used by the AMR-lab at the Technical University of Delft (TU Delft) to create a spatial-temporal 2D/3D occupancy dataset with pseudo-ground-truth annotations. This new dataset focuses on crowded urban environments and aims to facilitate occupancy learning from the perspective of mobile robots. The proposed method in this thesis will be employed to generate occupancy predictions for all pedestrians within the dataset.

The UT Campus dataset [71] is a large-scale multiclass, multimodal dataset focused on crowded urban environments recorded at the University of Texas. The dataset is partially annotated and contains ground-truth 3D bounding boxes across 53 object classes, LiDAR terrain segmentations, and global pose information. Unlike the other datasets used in this thesis, the UT Campus dataset is not specifically designed for HMR or 3D pose estimation, presenting several challenges:

1. The dataset does not contain 2D bounding boxes, pose, or tracking annotations.
2. Only a limited number of LiDAR frames have annotated 3D annotations/bounding boxes.
3. Many pedestrians are (partially) occluded due to the crowded environments.
4. The LiDAR-image alignment is inconsistent, particularly in frames where the recording vehicle rotates quickly.

The first three challenges are addressed by generating pseudo-ground-truth annotations using various 2D and 3D detection models. However, the accumulated noise in these pseudo-ground-truth annotations, combined with the inconsistent LiDAR-image alignment, makes finding a feasible solution in the optimization space difficult. To enhance the robustness of the proposed optimization scheme in such situations, the next section describes a minor modification to the original method outlined in Chapter 3.

Experimental setup
To make the proposed method more resilient to noisy input features and reduce dependency on precise LiDAR-image alignment, the optimization scheme, after initial ICP alignment, has been divided into three parts: 1) scaling optimization ($\mathcal{L}_{scale}$), 2) 3D mesh optimization with root alignment between the 2D and projected 3D root joint ($\mathcal{L}_m$), and 3) image alignment ($\mathcal{L}_{img}$). Compared to the original optimization scheme proposed in Chapter 3, this updated version should be more versatile and robust against imprecise LiDAR-image alignment. Additionally, the updated framework can be configured to prioritize either image or LiDAR alignment, enhancing its adaptability..

Scaling optimization:
After the ICP alignment between the initial SMPL prediction and LiDAR data (see section 3.3) the mesh scaling within the 2D image space is infeasible. Therefore, in the first optimization step, we *only* optimize for the shape parameters $\vec{\beta}$, responsible for mesh scaling.

$$\mathcal{L}_{scale}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}^{algn}) = \mathcal{L}_J(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}^{algn}) + \lambda_\beta \mathcal{L}_\beta(\vec{\beta}) \tag{11}$$

Where $\mathcal{L}_J$ and $\mathcal{L}_\beta$ follow the definitions of equations 3 and 6. However, the estimated 2D joints J_{est} originally within $\mathcal{L}_J$ are translated by $t \in \mathbb{R}^2$ to align the root joint J_{est_0} with the projected 3D root joint J_{pro_0}.

$$t = J_{pro_0} - J_{est_0} \tag{12}$$

$$J_{est}^{algn} = J_{est} + t \tag{13}$$

3D mesh optimization:
The second optimization aims to align the 3D pose and shape with both the image- and LiDAR features by optimizing the body pose $\vec{\theta}_{body}$, shape $\vec{\beta}$, and global rotation $\vec{\theta}_{glob}$ parameters. The objective function is very similar to equation 2. However, J_{est}^{algn} is used instead of the J_{est} to make this optimization step independent of image-LiDAR alignment.

Image alignment:
The first two steps of the updated optimization scheme improve the 2D and 3D alignment of the SMPL prediction but are independent of image-LiDAR alignment. This final optimization step can *only* optimize the camera translation t_{cam} to improve the final image or LiDAR alignment, depending on the scaler weight λ_{3D}.

$$\mathcal{L}_{img}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}) = \mathcal{L}_J(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, K, J_{est}) + \lambda_{3D} \mathcal{L}_{3D}(\vec{\beta}, \vec{\theta}, \vec{t}_{cam}, P, V) \tag{14}$$

Qualitative results

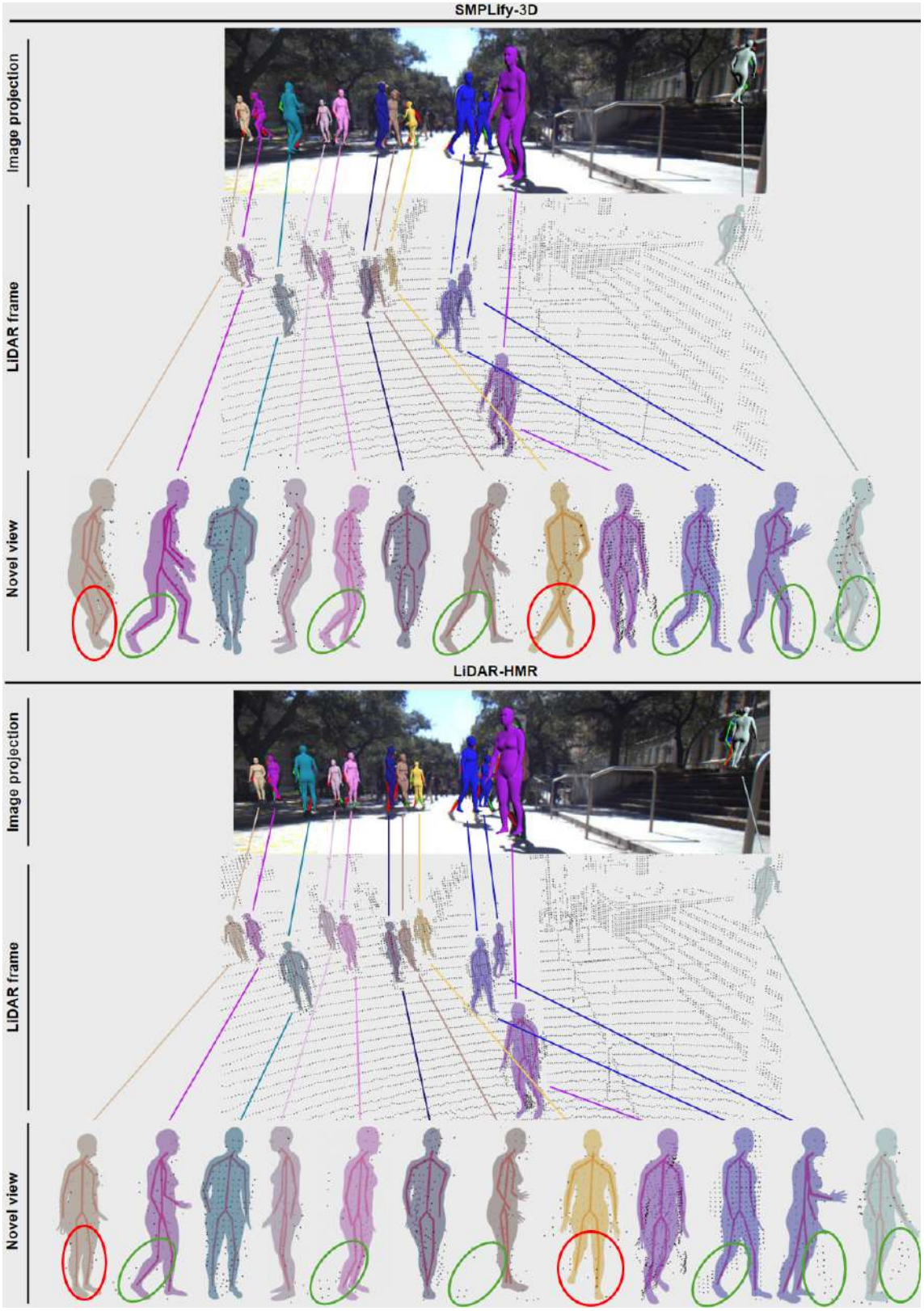


Figure 25: Qualitative comparison between SMPLify-3D and LiDAR-HMR [18] on the UT Campus dataset [71]. The ellipses indicate areas of notable difference: green ellipses show where SMPLify-3D outperforms LiDAR-HMR, while red ellipses highlight areas where LiDAR-HMR surpasses SMPLify-3D.

Qualitatively, SMPLify-3D outperforms LiDAR-HMR [18] in most instances within the UT Campus dataset [71], as shown in Figure 25. The performance differences are particularly noticeable in the 3D poses of the arms and legs, especially when limited LiDAR data is available (green ellipses). Additionally, SMPLify-3D's projected predictions align more accurately with the image. However, SMPLify-3D is not always superior. Instances with inaccurate 2D keypoint detections or initial SMPL predictions can lead to suboptimal solutions, as highlighted by the red ellipses.

5.3 Discussion

The experiments conducted in this chapter used HMR and 3D pose datasets with real LiDAR data [15, 17] to quantitatively compare the proposed method with other LiDAR-based HMR frameworks. Furthermore, the second part of this chapter shows qualitative results from data recorded within the AMR-lab and the UT Campus dataset [71].

The proposed HMR framework achieves competitive results on both the SLOPER4D [15] and HumanM3 [17] dataset compared to other LiDAR-based HMR frameworks, see table 9. Especially considering that the other methods are specifically trained on these datasets, while SMPLify-3D only required adjustments to several hyperparameters. Moreover, the HumanM3 dataset's multi-view setup demonstrates the versatility of our framework.

However, comparing mesh reconstruction quality between model-free and SMPL-based methods is challenging due to their differing mesh structures. The commonly used Per-Vertex-Error (PVE) can be misleading and may not accurately represent mesh reconstruction quality. To address this, [18] proposed the MPERE metric, however, this metric is biased towards SMPL-based methods. Therefore, comparing 3D pose accuracy is more representative, despite the model-free methods' ability to place joints anywhere in space, while SMPL-based models are constrained by the SMPL model/parameters. Compared to other SMPL-based methods, SMPLify-3D is the only framework that remains competitive on both datasets when evaluated alongside model-free methods. This may suggest that the current SMPL-compatible, LiDAR-based HMR frameworks are mainly focused on close-range scenarios or dense point cloud data. Despite the competitive results of SMPLify-3D, both the SLOPER4D and HumanM3 datasets exhibit limited variation in 3D poses compared to the 3DPW dataset used in Chapter 4, as shown in Figure 19, making the comparison biased towards these specific 3D poses.

The versatility of our method compared to other LiDAR-based methods is also evident in qualitative experiments on the AMR lab and UT Campus datasets. In commonly observed scenarios where the human subject is standing upright facing the camera/LiDAR, the performance of LiDAR-HMR and SMPLify-3D is comparable. However, in unique or uncommon 3D poses, LiDAR-HMR is unable to predict feasible results, making SMPLify-3D superior in these situations, see Figure 24.

Despite the demonstrated versatility of the proposed method, occlusions and misaligned input features can lead to inaccurate predictions. The proposed modification based on qualitative results from the UT Campus dataset aims to address these challenges, but there are no quantitative results that show the effectiveness of the updated optimization framework. Furthermore, the updated framework introduces more tunable weights in the objective function, complicating the tuning process.

6 Conclusion and Future works

6.1 Conclusion

This work introduced SMPLify-3D, a Human Mesh Recovery (HMR) framework that combines the richness of image data with the depth information from sparse LiDAR data to enhance the SMPL predictions made by purely image-based HMR frameworks. By leveraging multi-modal data, SMPLify-3D addresses common issues in image-based HMR, such as 3D pose inaccuracies and mesh placement errors due to depth ambiguity. The proposed framework consists of three main steps: 1) defining the visibility of all body parts, 2) initial LiDAR-mesh alignment, and 3) mesh optimization (see Figure 8).

Compared to image-based HMR frameworks, SMPLify-3D significantly improves PVE, MPJPE, and PA-MPJPE scores by 45%, 54%, and 10% respectively on the 3DPW [66] dataset with simulated LiDAR data. Additionally, SMPLify-3D's performance is directly applicable in real-life scenarios, unlike image-based HMR frameworks that require root joint alignment. SMPLify-3D also achieves competitive/superior performance on the HumanM3 [17] and SLOPER4D [15] datasets, despite these LiDAR-based HMR frameworks being specifically trained on those datasets. This highlights the versatility of SMPLify-3D. Additionally, experiments with simulated LiDAR data have revealed the correlation between LiDAR density, noise, and the framework's performance.

We also demonstrated the feasibility of deploying SMPLify-3D on a real mobile robot using the UT Campus [71] dataset and an experiment in the AMR lab. These qualitative experiments showcased the superior versatility and performance of our framework compared to LiDAR-HMR.

While this work focuses on improving image-based HMR predictions, SMPLify-3D can also enhance the image and 3D alignment of SMPL predictions made by non-image-based HMR frameworks.

6.2 Future works

Despite the demonstrated versatility and competitive performance of SMPLify-3D across multiple datasets, there are several avenues for future research to further enhance performance. Currently, the framework defines body-part visibility based on the confidence scores of human pose estimation (HPE) networks. However, these confidence scores may not be a reliable indication of prediction accuracy/occlusion [27]. Misidentified body-part visibilities can lead to suboptimal solutions and interpenetration errors that propagate through the optimization scheme. Therefore, developing an improved method for determining body-part visibility could significantly boost performance.

Additionally, the proposed optimization scheme does not contain a temporal and spatial consistency term. This implies that currently, consecutive SMPL predictions for the same person can differ significantly in position and 3D pose. Especially if the input space of the mesh optimization is noisy. This leads to fluctuating accuracies and potentially infeasible predictions. Incorporating temporal and spatial consistency terms within the objective function could mitigate these issues and further enhance the framework's robustness.

References

[1] RenderPeople, 2020.

[2] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. Learning to Reconstruct People in Clothing from a Single RGB Camera. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1175–1186, 3 2019.

[3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. *ACM SIGGRAPH 2005 Papers*, 2005.

[4] Ainesh Bakshi, Piotr Indyk, Rajesh Jayaram, Sandeep Silwal, and Erik Waingarten. A Near-Linear Time Algorithm for the Chamfer Distance. *Advances in Neural Information Processing Systems 36*, 7 2023.

[5] Paul J Besl and Neil D McKay. A Method for Registration of 3D shapes. *Sensor fusion IV: control paradigms and data structures*, 1611:586–606, 1992.

[6] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A Synthetic Dataset of Bodies Exhibiting Detailed Lifelike Animated Motion. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[7] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. SMPLify: Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. *Computer Vision–ECCV 2016*, 2016.

[8] Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Yanjun Wang, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, Chen Change Loy, Lei Yang, and Ziwei Liu. SMPLer-X: Scaling Up Expressive Human Pose and Shape Estimation. *arXiv preprint arXiv:2309.17448*, 9 2023.

[9] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, 1 2011.

[10] Junuk Cha, Muhammad Saqlain, GeonU Kim, Mingyu Shin, and Seungryul Baek. Multi-Person 3D Pose and Shape Estimation via Inverse Kinematics and Refinement. *European Conference on Computer Vision*, 10 2022.

[11] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*, 6 2014.

[12] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, 8 2020.

[13] Tabitha S. Combs, Laura S. Sandt, Michael P. Clamann, and Noreen C. McDonald. Automated Vehicles and Pedestrian Safety: Exploring the Promise and Limits of Pedestrian Detection. *American Journal of Preventive Medicine*, 56(1):1–7, 1 2019.

[14] Peishan Cong, Xinge Zhu, and Yuexin Ma. Input-Output Balanced Framework for Long-Tailed Lidar Semantic Segmentation. *IEEE International Conference on Multimedia and Expo*, 2021.

[15] Yudi Dai, Yitai Lin, Xiping Lin, Chenglu Wen, Lan Xu, Hongwei Yi, Siqi Shen, Yuexin Ma, and Cheng Wang. SLOPER4D: A Scene-Aware Dataset for Global 4D Human Pose Estimation in Urban Environments. *IEEE/CVF conference on computer vision and pattern recognition*, 3 2023.

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. *Conference on robot learning*, 2017.

[17] Bohao Fan, Siqi Wang, Wenxuan Guo, Wenzhao Zheng, Jianjiang Feng, and Jie Zhou. Human-M3: A Multi-view Multi-modal Dataset for 3D Human Pose Estimation in Outdoor Scenes. *arXiv preprint arXiv:2308.00628*, 8 2023.

[18] Bohao Fan, Wenzhao Zheng, Jianjiang Feng, and Jie Zhou. LiDAR-HMR: 3D Human Mesh Recovery from LiDAR. *arXiv preprint arXiv:2311.11971*, 11 2023.

[19] Mihai Fieraru, Mihai Zanfir, Silviu Cristian Pirlea, Vlad Olaru, and Cristian Sminchisescu. AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training. *IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *IEEE conference on computer vision and pattern recognition*, 2012.

[21] T A Gooley and H H Barrett. Evaluation of Statistical Methods of Image Reconstruction Through ROC Analysis. *IEEE transactions on medical imaging*, 11(2):276–283, 1992.

[22] Shanyan Guan, Jingwei Xu, Michelle Zhang He, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Out-of-Domain Human Mesh Reconstruction via Dynamic Bilevel Online Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5070–5086, 4 2022.

[23] Heinrich Hey and Werner Purgathofer. Occlusion Culling Methods. *Eurographics*, 2001.

[24] Clint Heyer. Human-Robot Interaction and Future Industrial Robotics Applications. *International conference on intelligent robots and systems*, 2010.

[25] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE transactions on pattern analysis and machine intelligence*, 2014.

[26] Haiyong Jiang, Jianfei Cai, and Jianmin Zheng. Skeleton-Aware 3D Human Shape Reconstruction From Point Clouds. *IEEE/CVF International Conference on Computer Vision*, 2019.

[27] Zhongyu Jiang, Haorui Ji, Cheng-Yen Yang, and Jenq-Neng Hwang. 2D Human Pose Estimation Calibration and Keypoint Visibility Classification. *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3 2024.

[28] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar Fine-Tuning for 3D Human Model Fitting Towards In-the-Wild 3D Human Pose Estimation. *International Conference on 3D Vision (3DV)*, 4 2021.

[29] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. HMR: End-to-end Recovery of Human Shape and Pose. *IEEE conference on computer vision and pattern recognition*, 12 2017.

[30] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video Inference for Human Body Pose and Shape Estimation. *IEEE/CVF conference on computer vision and pattern recognition*, 12 2019.

[31] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. PARE: Part Attention Regressor for 3D Human Body Estimation. *IEEE/CVF International Conference on Computer Vision*, 2021.

[32] Muhammed Kocabas, Chun-Hao P Huang, Joachim Tesch, Lea Müller, Otmar Hilliges, and Michael J Black. SPEC: Seeing People in the Wild with an Estimated Camera. *IEEE/CVF International Conference on Computer Vision*, 2021.

[33] Muhammed Kocabas, Ye Yuan, Pavlo Molchanov, Yunrong Guo, Michael J. Black, Otmar Hilliges, Jan Kautz, and Umar Iqbal. PACE: Human and Camera Motion Estimation from in-the-wild Videos. *arXiv preprint arXiv:2310.13768*, 10 2023.

[34] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. SPIN: Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. *IEEE/CVF international conference on computer vision*, 9 2019.

[35] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining Implicit Function Learning and Parametric Models for 3D Human Reconstruction. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK*, 2020.

[36] Jacob Lambert, Alexander Carballo, Abraham Monrroy Cano, Patiphon Narksri, David Wong, Eijiro Takeuchi, and Kazuya Takeda. Performance Analysis of 10 Models of 3D LiDARs for Automated Driving. *IEEE Access*, 8:131699–131722, 2020.

[37] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, and Michael J Black. Unite the People: Closing the Loop Between 3D and 2D Human Representations. *IEEE conference on computer vision and pattern recognition*, 2017.

[38] Jialian Li, Jingyi Zhang, Zhiyong Wang, Siqi Shen, Chenglu Wen, Yuexin Ma, Lan Xu, Jingyi Yu, and Cheng Wang. LiDARCap: Long-range Marker-less 3D Human Motion Capture with LiDAR Point Clouds. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[39] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A Hybrid Analytical-Neural Inverse Kinematics Solution for 3D Human Pose and Shape Estimation. *IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[40] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. FLAME: Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics*, 36(6), 11 2017.

[41] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying Location Information in Full Frames into Human Pose and Shape Estimation. *European Conference on Computer Vision*, 7 2022.

[42] Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-Stage 3D Whole-Body Mesh Recovery with Component Aware Transformer. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[43] Guanze Liu, Yu Rong, and Lu Sheng. VoteHMR: Occlusion-Aware Voting Network for Robust 3D Human Mesh Recovery from Partial Point Clouds. *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, pages 955–964, 10 2021.

[44] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6), 11 2015.

[45] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426*, 2 2018.

[46] Cade Metz. What's the future for A.I.? Technical report, 2023.

[47] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map. *IEEE conference on computer vision and pattern Recognition*, 2018.

[48] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. NeuralAnnot: Neural Annotator for 3D Human Mesh Training Sets. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[49] Lea Müller, Vickie Ye, Georgios Pavlakos, Michael Black, and Angjoo Kanazawa. Generative Proxemics: A Prior for 3D Social Interaction from Images. *arXiv preprint arXiv:2306.09337*, 6 2023.

[50] Thomas Alexander Sick Nielsen and Sonja Haustein. On sceptics and enthusiasts: What are the expectations towards self-driving cars? *Transport Policy*, 66:49–55, 8 2018.

[51] Ahmed A A Osman, Timo Bolkart, and Michael J Black. STAR: Sparse Trained Articulated Human Body Regressor. *Computer Vision–ECCV 2020*, 2020.

[52] Xueni Pan and Antonia F.de C. Hamilton. Why and how to use virtual reality to study human social interaction: The challenges of exploring a new research landscape. *British Journal of Psychology*, 109(3):395–417, 8 2018.

[53] Priyanka Patel, Chun-Hao P. Huang, Joachim Tesch, David T. Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in Geography Optimized for Regression Analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4 2021.

[54] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A A Osman, Dimitrios Tzionas, and Michael J Black. SMPL-X, Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[55] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. HuMoR: 3D Human Motion Model for Robust Pose Estimation. *IEEE/CVF international conference on computer vision*, 2021.

[56] Yiming Ren, Xiao Han, Chengfeng Zhao, Jingya Wang, Lan Xu, Jingyi Yu, and Yuexin Ma. LiveHPS: LiDAR-based Scene-level Human Pose and Shape Estimation in Free Environment. *arXiv preprint arXiv:2402.17171*, 2024.

[57] Javier Romero, Dimitrios Tzionas, and Michael J. Black. MANO, Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6), 11 2017.

[58] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. *IEEE/CVF international conference on computer vision*, 2019.

[59] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 1 2016.

[60] Karthik Shetty, Annette Birkhold, Srikrishna Jaganathan, Norbert Strobel, Markus Kowarschik, Andreas Maier, and Bernhard Egger. PLIKS: A Pseudo-Linear Inverse Kinematic Solver for 3D Human Body Estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[61] Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J. Black. WHAM: Reconstructing World-grounded Humans with Accurate 3D Motion. *arXiv preprint arXiv:2312.07531*, 12 2023.

[62] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. *IEEE/CVF conference on computer vision and pattern recognition*, 2023.

[63] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, One-stage, Regression of Multiple 3D People. *IEEE/CVF international conference on computer vision*, 2021.

[64] Yu Sun, Qian Bao, Wu Liu, Tao Mei, and Michael J. Black. TRACE: 5D Temporal Regression of Avatars with Dynamic Cameras in 3D Environments. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6 2023.

[65] Shixiang Tang, Cheng Chen, Qingsong Xie, Meilin Chen, Yizhou Wang, Yuanzheng Ci, Lei Bai, Feng Zhu, Haiyang Yang, Li Yi, Rui Zhao, and Wanli Ouyang. HumanBench: Towards General Human-centric Perception with Projector Assisted Pretraining. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[66] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. *European conference on computer vision (ECCV)*, 2018.

[67] Ze Yang, Shenlong Wang, Sivabalan Manivasagam, Zeng Huang, Wei-Chiu Ma, Xinchen Yan, Ersin Yumer, and Raquel Urtasun. S3: Neural Shape, Skeleton, and Skinning Fields for 3D Human Modeling. *IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[68] Zhitao Yang, Zhongang Cai, Haiyi Mei, Shuai Liu, Zhaoxi Chen, Weiye Xiao, Yukun Wei, Zhongfei Qing, Chen Wei, Bo Dai, Wayne Wu, Chen Qian, Dahua Lin, Ziwei Liu, and Lei Yang. SynBody: Synthetic Dataset with Layered Human Models for 3D Human Perception and Modeling. *arXiv preprint arXiv:2303.17368*, 3 2023.

[69] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. Generating Holistic 3D Human Motion from Speech. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[70] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. MubyNet: Deep Network for the Integrated 3D Sensing of Multiple People in Natural Images. *Advances in neural information processing systems*, 2018.

[71] Arthur Zhang, Chaitanya Eranki, Zhang Christina, Ji-Hwan Park, Raymond Hong, Pranav Kalyani, Lochana Kalyanaraman, Arsh Gamare, Arnav Bagad, Maria Esteva, and Joydeep Biswas. Towards Robust Robot 3D Perception in Urban Environments: The UT Campus Object Datase. *IEEE Transactions on Robotics*, 2024.

[72] Hansong Zhang and Kenneth E Hoff III. Fast Backface Culling Using Normal Masks. *Symposium on Interactive 3D graphics*, 1997.

[73] Siqi Zhang, Chaofang Wang, Wenlong Dong, and Bin Fan. A Survey on Depth Ambiguity of 3D Human Pose Estimation. *Applied Sciences (Switzerland)*, 12(20), 10 2022.

[74] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and ? Siyu Tang. EgoBody: Human Body Shape and Motion of Interacting People from Head-Mounted Devices. *European Conference on Computer Vision*, 2022.

[75] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul L. Rosin, Zixi Cai, Han Xi, Dingcheng Yang, Hao-Zhi Huang, and Shi-Min Hu. Pose2Seg: Detection Free Human Instance Segmentation. *IEEE/CVF conference on computer vision and pattern recognition*, 3 2018.

[76] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. MotionBERT: A Unified Perspective on Learning Human Motion Representations. *IEEE/CVF International Conference on Computer Vision*, 2023.