

MobileOcc: A Human-Aware Semantic Occupancy Dataset for Mobile Robots

Junseo Kim^{*1}, Guido Dumont^{*1}, Xinyu Gao^{*1}, Gang Chen^{*1},
Holger Caesar¹, and Javier Alonso-Mora¹

Delft University of Technology, Delft, Mekelweg 2, 2628 CD Delft, The Netherlands

Abstract. Dense 3D semantic occupancy perception is critical for mobile robots operating in pedestrian-rich environments, yet it remains underexplored compared to its application in autonomous driving. To address this gap, we present MobileOcc, a semantic occupancy dataset for mobile robots operating in crowded human environments. Our dataset is built using an annotation pipeline that incorporates static object occupancy annotations and a novel mesh optimization framework explicitly designed for human occupancy modeling. It reconstructs deformable human geometry from 2D images, then refines and optimizes it using associated LiDAR point data. Using MobileOcc, we establish benchmarks for two tasks: i) Occupancy prediction and ii) Pedestrian velocity prediction, using different methods, including monocular, stereo, and panoptic occupancy, with metrics and baseline implementations for reproducible comparison. Beyond occupancy prediction, we further assess our annotation method on 3D human pose estimation datasets. Results demonstrate that our method exhibits robust performance across different datasets. Our code and dataset are released at https://autonomousrobots.nl/paper_websites/mobileocc

Keywords: 3D semantic occupancy · Mobile robot · Occupancy prediction

1 Introduction

Semantic occupancy prediction [21, 30, 36, 46, 49] has emerged as a key perception component in modern navigation systems. It predicts a 3D field representing free, occupied, and unknown space, along with semantic labels, directly from image inputs. This enables the construction of a differentiable perception model that can be seamlessly integrated into downstream planning tasks. However, existing datasets [29, 45, 48] for training semantic occupancy prediction models are primarily developed for autonomous driving scenarios and focus on rigid objects such as roads, buildings, and vehicles. In contrast, non-rigid objects, such as humans, appear less frequently and are typically modeled by aggregating multi-view points into rigid bodies, neglecting their deformable nature.

For mobile robots operating in human-centered environments, this limitation is critical. Humans are among the most frequently encountered entities in navigation, and accurate occupancy prediction must capture their expressive geometry.

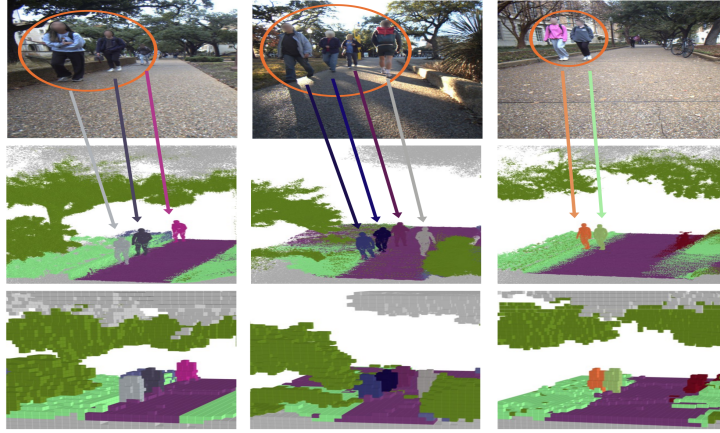


Fig. 1: Qualitative results across occupancy resolutions. Top: input image from UT Campus Object Dataset (CODa) [55]. Middle: our semantic occupancy label at fine resolution (0.02 m). Bottom: semantic occupancy at coarse resolution (0.2 m). Gray voxels denote unknown regions, whereas free space is not shown for clarity.

This is essential for socially aware navigation in dense crowds, interaction-centric navigation among pedestrians, and physical human–robot interaction requiring limb-level, pose-dependent representations.

In this paper, we present MobileOcc, a new semantic occupancy dataset designed for human-populated environments. The dataset is annotated using a proposed pipeline that fuses synchronized image and LiDAR data to generate occupancy labels, with a particular focus on accurate human representation. The pipeline produces occupancy annotations for static objects and for free and unknown space by fusing accumulated and filtered semantic LiDAR points within an OctoMap framework [19]. For human occupancy modeling, we introduce a mesh optimization method that first detects humans in images to estimate their initial pose and mesh, and then optimizes the mesh by fusing corresponding LiDAR points. Compared with image-only human mesh recovery (HMR) methods, which suffer from depth and scale ambiguities, and LiDAR-only approaches, which are limited by sparse points, missed detections, and pose uncertainty in cluttered or distant scenes [15, 27, 51], our fusion-based method produces robust and accurate human meshes by leveraging the strengths of both modalities.

The contributions of this paper are as follows:

- We introduce MobileOcc, a 3D semantic occupancy prediction dataset built on raw sensor data from UT Campus Object Dataset (CODa) [55] for mobile robots navigating among near-field pedestrians.
- We develop an annotation pipeline that generates occupancy annotations from synchronized RGB images and dense LiDAR data. In particular, the pipeline leverages both image data and LiDAR point clouds to produce accurate representations of human occupancy.

- We establish MobileOcc as a benchmark suite by evaluating existing occupancy frameworks on two tasks: i) semantic occupancy prediction and ii) pedestrian velocity prediction, with baselines across monocular, stereo, and panoptic occupancy methods.

2 Related Work

2.1 3D Occupancy Prediction Datasets

3D occupancy prediction is important for navigation and is widely used in autonomous driving. SemanticKITTI [1] introduced an occupancy perception dataset based on the KITTI dataset, but it does not account for moving objects and has limited scale and diversity. Subsequent datasets such as Occ3D [45], OpenOccupancy [48], and SSCBench [29] provide larger-scale, densely annotated voxel labels on nuScenes [4], Waymo [42], and KITTI-360 [1]. In off-road settings, WildOcc [54] extends RELLIS 3D [23] with dense semantic occupancy for unstructured environments. Tab. 1 summarizes key differences. These datasets typically model moving vehicles as rigid bodies and accumulate points using pre-labeled 3D bounding-box poses. However, humans are non-rigid and poorly represented in these datasets. For mobile robots, humans are among the most common dynamic objects encountered, and a dataset that accurately models human motion remains needed. Moreover, at fine resolution (Fig. 1), mesh-based labels preserve limb-level geometry that rigid bounding-box approximations cannot capture, which is critical for navigation in tight crowd gaps.

Table 1: Comparison with existing 3D occupancy datasets. MobileOcc is the first to provide non-rigid human occupancy with per-instance tracking for mobile robot navigation.

Dataset	Domain	Human Model	Instance Track	Non-rigid
SemanticKITTI [1]	Driving	–	✗	✗
Occ3D [45]	Driving	Rigid bbox	✗	✗
OpenOccupancy [48]	Driving	Rigid bbox	✗	✗
SSCBench [29]	Driving	Rigid bbox	✗	✗
OpenScene [10]	Driving	Rigid bbox	✓	✗
WildOcc [54]	Off-road	–	✗	✗
MobileOcc (Ours)	Pedestrian	SMPL mesh	✓	✓

2.2 3D Occupancy Prediction Methods

Unlike traditional filtering-based methods that fuse 3D points to generate an occupancy map [7], occupancy prediction learns to map 2D images to voxel-level semantic occupancy [46]. Camera-only methods such as VoxFormer [30] and OccDepth [36] infer depth cues and decode a dense 3D volume. Surround-view approaches [21, 46, 49] aggregate multi-camera features to improve coverage.

Efficient variants such as FlashOcc [52] and Panoptic-FlashOcc [53] keep features in BEV and lift logits to 3D. Our setting is complementary: we focus on mobile robots in human-rich environments and evaluate such frameworks under non-rigid human motion.

2.3 Image-Based Human Mesh Recovery

Estimating a 3D human body mesh from a single RGB image has been widely studied using parametric models such as SMPL [35]. Regression-based methods [24, 26, 31] directly predict shape and pose parameters, while alternative designs lift skeletons or heatmaps to meshes [9, 38], or use transformer regressors [32, 33]. However, purely image-based pipelines suffer from depth ambiguity. To reduce this uncertainty, LiDAR-based methods [15, 27, 56] fit SMPL meshes to point clouds, while S3 [51] uses implicit fields. LiDAR-only methods can be brittle when human points overlap with nearby objects or become sparse at long distances. We present a training-free method that first estimates an initial mesh from an image, then refines it by fusing LiDAR data, leveraging both modalities and generalizing directly across datasets.

3 MobileOcc Dataset Building

Our dataset is constructed by annotating the image and LiDAR data from the CODa dataset [55] using the proposed annotation pipeline. This section introduces the pipeline by first providing a general overview of the workflow and then detailing its key components.

3.1 Annotation Pipeline

Our general annotation pipeline is shown in Fig. 2. Given synchronized image streams and LiDAR along a robot trajectory, we first run OC-SORT [5] with YOLOX [16] detection to obtain tracked 2D pedestrian bounding boxes with IDs. For each tracked person, we estimate 2D pose with keypoint confidence using VITPose [50], obtain instance masks via Mask2Former [8], and associate each mask with the LiDAR point cloud. We apply semantic segmentation to assign Cityscapes [11] class labels for static background elements. All preprocessed data are time-stamped and co-referenced across modalities. With these preprocessing steps, we perform human mesh optimization (Sec. 3.2) and static mapping (Sec. 3.3), then fuse them into a 3D occupancy representation (Sec. 3.4).

3.2 Human Mesh Optimization

We propose an optimization-based framework that fuses LiDAR points with monocular images to improve 3D human mesh estimation. Our approach first obtains an initial parametric mesh (SMPL) from the image [35], then refines it in three stages: i) filtering out occluded body parts using 2D keypoint cues, ii)

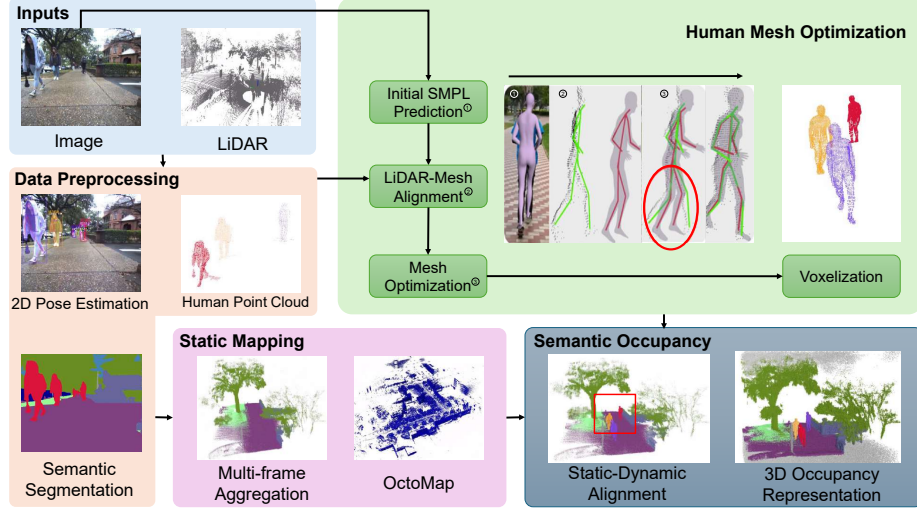


Fig. 2: Overview of MobileOcc pipeline. The pipeline consists of: Data preprocessing, Static mapping, Human mesh optimization, and 3D occupancy representation generation. The colors of the voxels follow the labeling scheme used in the Cityscapes dataset [11]. Human instances are assigned colors, and unknown regions are shown in gray. In human mesh optimization, the green and red skeletons represent the ground truth and the predicted 3D pose of the mesh, respectively.

performing a coarse mesh alignment to LiDAR via iterative closest point (ICP), and iii) optimizing the SMPL pose/shape parameters to fit the image and LiDAR observations jointly.

We use the following notation throughout the paper:

- **SMPL parameters:** pose $\theta = \{\theta_{\text{glob}}, \theta_{\text{body}}\}$, shape β , and camera translation $\mathbf{t}_{\text{cam}} \in \mathbb{R}^3$.
- **Camera:** intrinsics $\mathbf{K}$.
- **Mesh:** $\mathcal{M}$ with vertex matrix $\mathbf{M}(\beta, \theta) \in \mathbb{R}^{6890 \times 3}$; visible subset $\mathcal{V} \subset \mathcal{M}$ with $n = |\mathcal{V}|$.
- **LiDAR:** point cloud $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^m$ with $m = |\mathcal{P}|$.
- **Operators:** projection $\Pi_{\mathbf{K}}(\cdot)$, global rotation $\mathbf{R}(\theta)$, and SMPL joints $\mathbf{J}_i(\beta)$.

Initial SMPL Prediction We obtain an initial SMPL estimate using CLIFF [31], a top-down HMR regressor that predicts shape β^0 , pose θ^0 , and translation $\mathbf{t}^0$ from a person crop conditioned on its full-image bounding box location. This provides a reasonable initial pose and location in the camera frame, but like all monocular methods, it suffers from depth ambiguity, which our subsequent steps correct.

LiDAR-Mesh Alignment Before optimizing the human mesh provided by CLIFF, we first perform visibility filtering to identify the mesh vertices visible

in the LiDAR data. We then coarsely register the visible mesh vertices $\mathcal{V}$ to the LiDAR points $\mathcal{P}$ via ICP [2], estimating a rigid transform $\mathbf{T} \in SE(3)$ initialized from $(\boldsymbol{\theta}_{\text{glob}}, \mathbf{t}_{\text{cam}})$. Applying $\mathbf{T}$ yields an updated global pose $(\boldsymbol{\theta}'_{\text{glob}}, \mathbf{t}'_{\text{cam}})$ that places the mesh correctly in 3D space. However, since ICP is rigid and does not alter internal joint angles, the pose can become inconsistent with the image. Our final optimization step resolves these inconsistencies by adjusting pose and shape using both image and LiDAR cues.

Mesh Optimization To resolve the remaining pose inconsistencies highlighted in Fig. 2 (red ellipse), we refine all SMPL parameters $(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{t}_{\text{cam}})$ to better fit both the 2D and 3D data. Inspired by SMPLify [3], we formulate this as minimizing an objective function

$$\begin{aligned} \mathcal{L}_{\text{total}}(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{t}_{\text{cam}}) = & \mathcal{L}_J + \lambda_{3D} \mathcal{L}_{3D} + \lambda_{\theta} \mathcal{L}_{\theta} + \lambda_a \mathcal{L}_a \\ & + \lambda_{\beta} \mathcal{L}_{\beta} + \lambda_{\text{occ}} \mathcal{L}_{\theta, \text{occ}} \end{aligned} \quad (1)$$

that contains terms for image alignment, 3D alignment, and regularization priors with scalar weights λ :

- **2D joint reprojection** ($\mathcal{L}_J$): Penalizes the distance between the projected 3D joints of the mesh and the detected 2D keypoints in the image,

$$\mathcal{L}_J = \sum_i w_i \rho \left(\Pi_{\mathbf{K}}(\mathbf{R}(\boldsymbol{\theta}) \mathbf{J}_i(\boldsymbol{\beta})) + \mathbf{t}_{\text{cam}} - \mathbf{J}_i^{\text{est}} \right), \quad (2)$$

where w_i is the keypoint confidence and $\rho(\cdot)$ is the Geman–McClure robust penalty [17].

- **3D LiDAR alignment** ($\mathcal{L}_{3D}$): Minimizes the distance between visible mesh vertices $\mathcal{V}$ and the LiDAR point cloud $\mathcal{P}$ using a symmetric, size-normalized Chamfer distance:

$$\mathcal{L}_{3D} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{v} \in \mathcal{V}} \min_{\mathbf{p} \in \mathcal{P}} \|\mathbf{v} - \mathbf{p}\|_2^2 + \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \min_{\mathbf{v} \in \mathcal{V}} \|\mathbf{p} - \mathbf{v}\|_2^2. \quad (3)$$

Optimizing over $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ lets this term correct local pose/shape errors that pure 2D fitting cannot.

- **Pose prior** ($\mathcal{L}_{\theta}$): The SMPLify MoG pose prior [3] discourages implausible joint configurations,

$$\mathcal{L}_{\theta} = -\log \sum_{j=1}^N g_j \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta, j}, \boldsymbol{\Sigma}_{\theta, j}). \quad (4)$$

- **Anti-hyperextension prior** ($\mathcal{L}_a$): Discourages unnatural positive bending at elbows/knees (as in [3]),

$$\mathcal{L}_a = \sum_{k \in \{\text{knees}, \text{elbows}\}} \exp(\theta_k). \quad (5)$$

- **Shape prior** ($\mathcal{L}_\beta$): A quadratic prior on shape parameters keeps β within the SMPL space [35],

$$\mathcal{L}_\beta = \beta^\top \Sigma_\beta^{-1} \beta. \quad (6)$$

- **Occlusion-aware pose consistency prior** ($\mathcal{L}_{\theta, \text{occ}}$): For joints flagged invisible by the visibility filter, we regularize deviation from the initialization to prevent unrealistic drift. Let $\mathcal{I}$ be the set of occluded joints with initial and current unit quaternions $\{\mathbf{q}_i^{(0)}, \mathbf{q}_i\}$; we use

$$\mathcal{L}_{\theta, \text{occ}} = \sum_{i \in \mathcal{I}} \left(1 - \langle \mathbf{q}_i^{(0)}, \mathbf{q}_i \rangle^2\right). \quad (7)$$

We optimize $\mathcal{L}_{\text{total}}$ with respect to $(\beta, \theta, \mathbf{t}_{\text{cam}})$ using gradient descent in PyTorch. The result is a refined set $(\hat{\beta}, \hat{\theta}, \mathbf{t}_{\text{cam}}^*)$ that balances fitting the image (2D joints) and fitting the LiDAR (3D points), while respecting human shape and pose priors. In contrast to SMPLify [3], we omit the interpenetration term and instead add $\mathcal{L}_{3D}$ (LiDAR-mesh Chamfer) and $\mathcal{L}_{\theta, \text{occ}}$ (occlusion-aware pose consistency), while keeping $\mathcal{L}_J$, $\mathcal{L}_\theta$, $\mathcal{L}_a$, $\mathcal{L}_\beta$. We drop interpenetration due to its high cost and limited impact on final accuracy [26]. Our occlusion-aware visibility and pose-consistency terms help keep occluded limbs in reasonable positions without explicit handling of interpenetration. Moreover, since we voxelize occupancy from the final mesh surface (i.e., the exterior boundary), interpenetration primarily results in rare internal self-overlaps and has minimal effect on the occupied/free voxel boundary.

3.3 Static Map Generation

We build a high-resolution static semantic map by fusing multi-frame LiDAR geometry with camera-derived semantics, while explicitly filtering dynamic pedestrians, including those outside the camera frustum.

LiDAR-based Detection To avoid moving pedestrians being imprinted into the background, we run a LiDAR 3D detector per sweep and use its pedestrian outputs, together with the image-based pedestrian detection result, as a mask during static mapping. Points that fall within these masks are excluded from static map generation. This stage neither changes semantic labels nor trains any model, and its sole purpose is to suppress pedestrian evidence before building the static map.

Multi-frame Aggregation We fuse LiDAR geometry and camera semantics over time in a global frame. Each sweep is pose-compensated, and image semantics are lifted to the corresponding 3D points, which are then inserted into a voxel map that maintains compact per-voxel label counts. As frames accumulate, each voxel label is updated through max-voting, and geometric sanity checks (e.g., ground consistency) reject outliers. After processing the sequence,

the final static map is obtained by assigning a semantic label to each voxel, yielding a human-free background that we later combine with human meshes to obtain human-aware occupancy.

3.4 3D Occupancy Representation

Static-dynamic Alignment We merge the human-free static map (Sec. 3.3) with pedestrian evidence (Sec. 3.2) in a shared robot-local frame as shown in Fig. 2. For each frame, we transform the current static points and rasterized meshes into a common local coordinate frame using the synchronized robot pose, then voxelize them. Static voxels inherit Cityscapes [11] semantics from RGB points. Pedestrian voxels carry per-instance identifiers, as we assign each tracked pedestrian a unique ID label.

Label Assignment After alignment, we instantiate voxel grids in the robot frame with a user-specified resolution (up to 0.02 m) and assign labels in a priority order: dynamic humans take precedence over static background, so pedestrians do not get imprinted into the background during fusion. If no pedestrian points are present, we assign static semantics using a majority-label-wins rule to avoid label oscillation when many points land in the same voxel. Remaining cells are completed using an OctoMap query [19], which contributes free/unknown states so that occupied space is supported by explicit geometry. This produces an aligned voxel grid in which static semantics and dynamic human occupancy coexist, and all remaining space is consistently categorized as either free or unknown.

We output our final data in the NuScenes [4] dataset format to ensure easier integration and consistency with existing tools.

4 Results

In this section, we present a comprehensive evaluation of our approach. We first evaluate human mesh optimization on three datasets. We then validate the quality of the generated occupancy labels against available ground-truth annotations and quantify the robustness of upstream detection. Finally, we train representative baselines on MobileOcc and report performance on two tasks: i) occupancy prediction and ii) pedestrian velocity prediction.

4.1 Human Mesh Evaluation

We evaluate our human mesh optimization method in 3DPW [47], SLOPER4D [12], and HumanM3 [14] datasets. We report four standard metrics: PVE (Per-Vertex Error, mm), the average vertex-level Euclidean distance; MPJPE (Mean Per-Joint Position Error, mm) over the dataset’s kinematic joint set; PA-MPJPE (Procrustes-Aligned MPJPE, mm), which isolates articulated pose quality from

Table 2: Simulated LiDAR comparison on 3DPW [47]. Video/Image methods use no depth and are reported with root alignment. Ours fuses simulated LiDAR without root alignment.

Modality	Method	PVE ↓	MPJPE ↓	PA-MPJPE ↓
Video	VIBE [25]	99.1	82.9	51.9
	DynaBOA [18]	82.0	65.5	40.4
	MotionBERT [57]	79.4	68.8	40.6
	WHAM-B [41]	71.0	59.4	37.2
	SMPLify [3]	106.8	–	–
	TRACE [44]	97.3	79.1	37.8
Image	SPIN [26]	96.9	59.2	–
	ROMP [43]	93.4	76.7	47.3
	HybrIK [28]	82.3	71.6	41.8
	CLIFF [31]	81.2	69.0	43.0
	Cha. [6]	76.3	66.0	39.0
	PLIKS [40]	73.3	60.5	38.5
LiDAR	Ours (Ouster-32)	73.2	57.0	47.7
	Ours (Ouster-64)	57.2	43.9	38.5
	Ours (Ouster-128)	50.5	39.1	35.1

global placement; and MPERE [15] (Mean Per-Edge Relative Error), the average relative error of mesh edge lengths. PVE and MPERE are reported only when the vertex ground truth is available.

Results on the 3DPW Dataset 3DPW [47] dataset is a popular in-the-wild dataset with 3D pose and shape ground truth. Since no LiDAR points are available, we add synthetic LiDAR sweeps at different densities in the camera frame. Specifically, we mimic Ouster-style sensors with vertical beams = 32, 64, 128 (higher → denser body coverage) and added Gaussian range perturbations and probabilistic point dropouts following the CARLA LiDAR noise model [13]. This allows us to i) isolate the effect of adding LiDAR points to image/video HMR; and ii) study how performance scales with sensor density.

Tab. 2 presents the results of our method compared with other image- and video-based HMR approaches evaluated on the 3DPW dataset. Since image- and video-based methods inherently lack depth, they are evaluated using root alignment [47], which translates the predicted pelvis to the ground-truth position. In contrast, our method instead uses simulated LiDAR on the human body to optimize position, pose, and shape directly in the sensor frame.

The results show that our method improves in all metrics without using root alignment, indicating better absolute placement (PVE/MPJPE) and pose accuracy (PA-MPJPE). Performance increases with denser LiDAR, which provides more reliable 3D correspondences.

Results on the SLOPER4D and HumanM3 Datasets We further evaluate our method on SLOPER4D [12] and HumanM3 [14] datasets, which are targeted for LiDAR-based 3D human pose estimation. SLOPER4D [12] has synchronized RGB image and LiDAR points, and ground-truth mesh vertices. An Ouster-OS1-128 LiDAR is used for data collection, and only one person appears in the

Table 3: Comparison on real LiDAR: SLOPER4D and HumanM3. Modality: RGB = image, LiDAR = point clouds, RGB+LiDAR = fused. The PA-MPJPE (mm) columns isolate the effect of our LiDAR optimization over the CLIFF initialization.

Method	SLOPER4D [12]				HumanM3 [14]	
	PVE ↓	MPJPE ↓	PA-MPJPE ↓	MPERE ↓	MPJPE ↓	PA-MPJPE ↓
<i>LiDAR-based</i>						
PRN [15]	–	57.0	–	–	82.2	–
V2V-PoseNet [37]	–	50.7	–	–	83.0	–
LiDAR-HMR [15]	51.9	51.0	–	0.094	77.6	–
LiDARCap [27]	148.1	158.3	–	0.050	175.8	–
SAHSR [22]	81.2	72.6	–	0.085	105.5	–
VoteHMR [34]	60.9	54.6	–	0.079	105.8	–
<i>RGB-based</i>						
CLIFF [31]	93.7	84.7	69.3	0.057	106.3	66.5
<i>RGB+LiDAR</i>						
Ours	64.4	55.8	43.5	0.060	83.9	58.1

dataset at a time. HumanM3 [14] has multiple humans at a distance, and a Livox Mid-100 LiDAR is used. It has ground-truth for 3D joints but not for the mesh.

On SLOPER4D, we report PVE, MPJPE, PA-MPJPE, and MPERE [15]. MPERE is computed for any method that outputs a mesh (SMPL-based or model-free that reconstructs surfaces). On HumanM3, vertex ground truth is unavailable, so we report only joint metrics (MPJPE and PA-MPJPE). At test time, we reuse the 3DPW pipeline: visibility filtering, single-sweep rigid pre-alignment, and joint refinement using RGB and LiDAR, all evaluated in the sensor frame.

We compare against LiDAR methods benchmarked on these two datasets and present additional results using our base image model, CLIFF [31]. As summarized in Tab. 3, our method achieves strong accuracy without any LiDAR-specific training. The PA-MPJPE columns directly isolate the effect of our LiDAR-guided optimization: comparing CLIFF (RGB) to Ours (RGB+LiDAR) shows substantial reductions on both SLOPER4D and HumanM3, confirming that most gains arise from test-time optimization with LiDAR constraints. Most competing methods are trained on LiDAR (and often dataset-specific) supervision, whereas our improvements arise purely from the optimization process at inference.

4.2 Occupancy Label Quality

Pointwise manual semantic annotation of 3D LiDAR data is prohibitively expensive, and CODa provides such labels for only a small portion of its sequences. We validate our voxel labels against CODa ground truth on 4 sequences (400 frames) and assess detection reliability across conditions; quantitative results are reported in Tab. 4.

■ pedestrian ■ car ■ other struct. ■ pole ■ road ■ terrain ■ truck
■ two-wheeler ■ vegetation

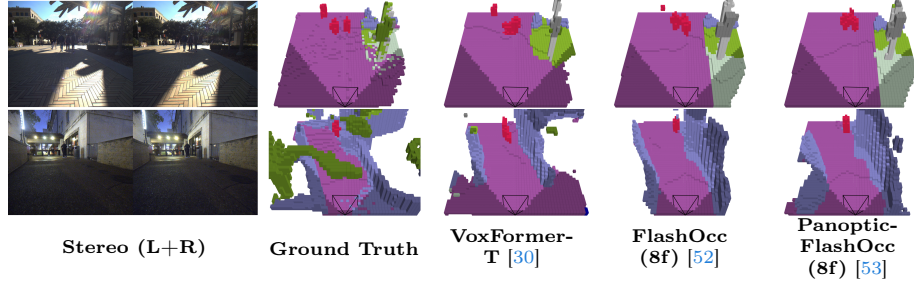


Fig. 3: Qualitative comparison of different baselines under various lighting conditions, including sunny, night, and cloudy scenes.

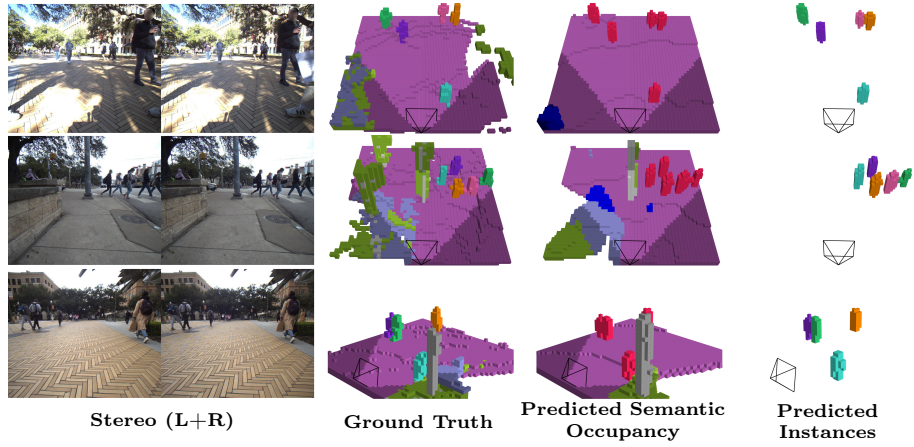


Fig. 4: Panoptic occupancy prediction performance using Panoptic-FlashOcc (8f) [53].

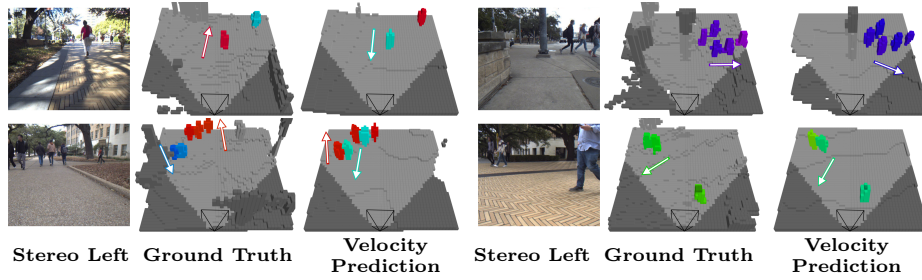


Fig. 5: Pedestrian velocity prediction using Panoptic-FlashOcc-vel (8f). The directions are shown in colors and arrows.

Static voxel-label quality We evaluate final static semantic voxel labels against CODa’s ground-truth semantic LiDAR point annotations for road, veg-

etation, and terrain. High GT coverage confirms that our voxel grid captures nearly all annotated geometry. The primary error source is taxonomy and domain mismatch between our Cityscapes-trained segmentation and CODa’s campus-outdoor categories; e.g., CODa’s “Short Vegetation” maps to Cityscapes’ *terrain* or *vegetation* depending on context, which mainly limits vegetation and terrain IoU.

Pedestrian occupancy quality We evaluate dynamic pedestrian voxel labels using CODa 3D pedestrian bounding boxes as proxy ground truth. Since CODa does not provide dense human-geometry annotations, bounding boxes serve as the best available reference; however, they are inherently conservative because they overcount occupied volume by including empty space around the body and by including ground-plane voxels at pedestrians’ feet. We therefore consider only camera-visible occupied voxels within each bbox volume and exclude the bottom 0.1 m of each bbox, which contains road/terrain voxels rather than body geometry. As shown in Tab. 4, precision is 84.4% and recall is 81.9%, confirming that our labels capture the majority of pedestrian geometry.

Pedestrian occupancy quality vs. LiDAR body GT. Bounding boxes overcount occupied volume, so we additionally validate the SMPL-derived voxels against a tighter reference built from raw LiDAR returns. For each pedestrian within the benchmark range, we extract the LiDAR points that fall inside CODa’s human-annotated 3D bounding box and voxelize them on the 0.2 m evaluation grid to obtain a per-pedestrian LiDAR body ground truth, a sampling of the sensor-facing body surface, restricted to the camera frustum to ensure a like-for-like comparison with our SMPL voxels. LiDAR returns lie on the visible body surface, whereas SMPL voxelization fills the body volume, so a LiDAR voxel naturally sits roughly half a body-depth (~ 0.19 m, measured) ahead of the corresponding SMPL voxel at the body center. We therefore evaluate at the standard ± 1 voxel-size center-to-center tolerance, with precision (SMPL $\rightarrow$ GT) and recall (GT $\rightarrow$ SMPL). Tab. 5 reports per-sequence results across 400 frames and 790 pedestrian instances. Precision reaches 95.5 and F1 92.5, substantially above the conservative bounding-box proxy, confirming that the pedestrian voxels closely match true body geometry.

Detection robustness To quantify how upstream detection errors propagate into occupancy labels, we perform manual quality assurance on 300 frames spanning crowded, uncrowded, and nighttime conditions (Tab. 4). Daytime detection achieves high recall and precision across both crowd densities. Nighttime recall decreases due to low illumination, while precision remains high, indicating that missed detections dominate false positives. These misses affect only the frames in which they occur and do not propagate into the static map, since pedestrian filtering already excludes detected humans from static fusion and human occupancy overrides static labels during alignment. We distinguish between daytime

Table 4: Occupancy label quality and upstream detection robustness. Top: voxel labels evaluated against CODa ground truth (4 sequences, 400 frames). Bottom: manual detection QA on 300 frames (T : correct; $M_{\text{vis}}/M_{\text{occ}}$: missed visible/occluded(counts); W : false positives).

Static Semantic Voxels (vs. GT LiDAR points)							
mIoU (road / veg. / terrain)	71.2 (88.8 / 68.0 / 57.0)						
GT point coverage	96.8						
Voxel-level accuracy	88.1						
Point-level accuracy	94.0						
Pedestrian Voxels (vs. GT 3D bounding boxes)							
Precision	84.4						
Recall	81.9						
F1	83.1						
Detection Robustness (manual QA, 300 frames)							
Condition	T	M_{vis}	M_{occ}	W	Recall	Precision	Miss $_{\text{vis}}\%$
Crowded	480	6	6	5	98.8	99.0	1.2
Uncrowded	191	1	2	0	99.5	100	0.5
Night	40	12	3	1	76.9	97.6	23.1

Table 5: Pedestrian voxel quality vs. LiDAR body GT, per CODa sequence (400 frames, 790 pedestrian instances).

Seq	7	11	16	18	All
Precision $\uparrow$	95.2	98.9	91.1	96.9	95.5
Recall $\uparrow$	83.6	87.8	95.6	92.6	89.9
F1 $\uparrow$	89.0	93.0	93.3	94.7	92.5

and nighttime sequences in the dataset, enabling users to select subsets that match their target deployment conditions.

5 Benchmark

5.1 Benchmark Details

We evaluate all methods at 0.2m voxel resolution over a grid spanning $x \in [0.4, 10.0]$ m, $y \in [-4.8, 4.8]$ m, $z \in [-1.0, 3.8]$ m, downsampled to 5 Hz. The training/validation split contains 92,303/24,208 frames (79/21%), of which 30,457/7,165 have pedestrian annotations. We use 10 semantic classes (one free-space, nine occupied), merging rare categories: bicycles, motorcycles, and scooters into two-wheeler, and several static structural categories into other-structure.

5.2 Baseline Implementations

Monocular BEV-based baselines. Monocular baselines include BEVDet4D [20] as a detection and velocity baseline, and FlashOcc [52] and Panoptic-FlashOcc [53] as semantic and panoptic 3D occupancy baselines. We use 8 historical frames

based on their public settings. The original BEVDet4D, FlashOcc, and Panoptic-FlashOcc models assume multi-view surround-camera inputs, whereas our dataset contains only front-view images. Therefore, we adapt them to a monocular setting by using only the left stereo camera to estimate depth. We also introduce Panoptic-FlashOcc-vel, which extends the detection head of Panoptic-FlashOcc to predict velocities (v_x, v_y) with an L1 loss for velocity supervision.

Stereo baseline: VoxFormer [30]. We adapt VoxFormer-T with 4 historical frames to our stereo setup, sampling frames alternately in time to match the temporal coverage of the 8-frame monocular baselines. Stage-1 takes voxelized stereo-matching depth as input and predicts a 3D occupancy volume at 0.4m resolution, which is then upsampled to our 0.2m target grid. We follow the public training settings of each baseline.

5.3 Benchmark Results

3D occupancy prediction. Tab. 6 and Fig. 3 compare the baselines on the semantic occupancy prediction task. FlashOcc (FO) achieves the highest mean IoU (mIoU) and outperforms the other methods on most classes. Panoptic-FlashOcc (PF) yields a minor mIoU drop relative to FO due to the multi-task trade-off. VoxFormer (VF) remains competitive on pedestrians, possibly because it benefits from stereo depth, whereas FlashOcc may inherit monocular pedestrian detection errors from BEVDet4D pretraining. Cars and trucks show lower IoU than in self-driving benchmarks, primarily because they are less common in our dataset.

Panoptic occupancy. We further evaluate the pedestrian detection and panoptic occupancy quality using BEVDet4D and Panoptic-FlashOcc. As shown in Tab. 7 and Fig. 4, BEVDet4D (BD) provides only pedestrian detections, while Panoptic-FlashOcc (PF) predicts voxel-level panoptic labels. Pedestrian Average Precision (AP^{Ped}) is evaluated across distance thresholds 0.1 m, 0.2 m, 0.5 m and 1 m on detected centers. In addition to standard panoptic metrics, we also report $PQ^\dagger$, a relaxed panoptic quality [39] that relaxes the strict IoU requirement for stuff classes and is better suited for vision-only occupancy prediction.

Pedestrian velocity prediction. Finally, we compare BEVDet4D and Panoptic-FlashOcc-vel (PF-vel) on the pedestrian velocity prediction task. Tab. 7 summarizes the absolute velocity error (AVE; m/s) over all ground truth pedestrian voxels (AVE-T), true-positive detections within 1 m (AVE-D), and correctly classified voxels occupied by pedestrians (AVE-O), together with mIoU. PF-vel achieves comparable velocity accuracy while maintaining voxel-level semantic occupancy. Qualitative examples in Fig. 5 show that PF-vel produces plausible velocities for most pedestrians but still exhibits typical failure cases, such as confusion between forward and backward walking directions, indicating

Table 6: 3D occupancy prediction performance. VF-T: VoxFormer-T, FO: FlashOcc, PF: Panoptic-FlashOcc. Best in **bold**. IoU denotes geometric IoU.

Method	IoU	mIoU	ped.	car	oth.str.	pole	road	terrain	truck	2-wh.	veg.
VF-T	57.81	24.89	32.79	1.39	28.47	7.08	70.90	23.57	5.23	28.04	26.52
FO (8f)	57.71	27.18	31.91	5.36	31.00	10.16	70.82	27.29	7.35	30.85	29.88
PF (8f)	57.83	26.64	32.45	3.66	31.79	9.96	70.35	25.90	5.87	30.99	28.83

Table 7: Panoptic occupancy and pedestrian velocity prediction. BD: BEVDet4D, PF: Panoptic-FlashOcc, PF-vel: Panoptic-FlashOcc-vel.

Method	Scene-level				Pedestrian			AP ^{Ped}	AVE (m/s)↓			
	PQ	PQ [†]	RQ	SQ	PQ ^{Ped}	RQ ^{Ped}	SQ ^{Ped}		T	D	O	mIoU↑
BD (8f)	—	—	—	—	—	—	—	41.7	1.00	0.36	—	—
PF (8f)	19.9	32.6	28.1	65.8	42.5	60.2	70.7	45.5	—	—	—	—
PF-vel (8f)	—	—	—	—	—	—	—	—	0.97	0.39	0.67	26.00

that there is remaining headroom for improving pedestrian velocity prediction in conjunction with panoptic occupancy.

6 Conclusion

This paper presents MobileOcc, a human-aware semantic occupancy dataset for mobile robots operating in pedestrian-dense, near-field environments. We provide baselines for occupancy prediction and pedestrian-velocity estimation across monocular, stereo, and panoptic settings. We further validate the human-mesh optimization component on established datasets, where adding LiDAR points for optimization substantially improves image-based methods and remains robust across them. Together, the dataset, pipeline, and benchmarks form a practical testbed for perception stacks that must jointly reason about humans, static objects, and free space to enable safe, precise navigation in crowds.

Limitations and future work. Currently, MobileOcc focuses on outdoor scenes, which may limit generalization across domains and conditions. Future extensions include broadening geographic and environmental diversity (e.g., indoor spaces and adverse weather). Additionally, our non-rigid model uses SMPL and does not capture carried or accompanying objects; we plan to extend the benchmark toward human-object occupancy to better handle such cases.

Acknowledgements

This work was supported by the European Union through ERC INTERACT, Grant 101041863. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9297–9307 (2019) [3](#)
2. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor fusion IV: control paradigms and data structures. vol. 1611, pp. 586–606. Spie (1992) [6](#)
3. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In: European conference on computer vision. pp. 561–578. Springer (2016) [6](#), [7](#), [9](#)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) [3](#), [8](#)
5. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9686–9696 (2023) [4](#)
6. Cha, J., Saqlain, M., Kim, G., Shin, M., Baek, S.: Multi-person 3d pose and shape estimation via inverse kinematics and refinement. In: European conference on computer vision. pp. 660–677. Springer (2022) [9](#)
7. Chen, G., Wang, Z., Dong, W., Alonso-Mora, J.: Particle-based instance-aware semantic occupancy mapping in dynamic environments. Trans. Rob. **41**, 1155–1171 (Jan 2025), <https://doi.org/10.1109/TR0.2025.3526084> [3](#)
8. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022) [4](#)
9. Choi, H., Moon, G., Lee, K.M.: Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In: European Conference on Computer Vision. pp. 769–787. Springer (2020) [4](#)
10. Contributors, O.: Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. <https://github.com/OpenDriveLab/OpenScene> (2023) [3](#)
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016) [4](#), [5](#), [8](#)
12. Dai, Y., Lin, Y., Lin, X., Wen, C., Xu, L., Yi, H., Shen, S., Ma, Y., Wang, C.: Sloper4d: A scene-aware dataset for global 4d human pose estimation in urban environments. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 682–692 (2023) [8](#), [9](#), [10](#)
13. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017) [9](#)
14. Fan, B., Wang, S., Guo, W., Zheng, W., Feng, J., Zhou, J.: Human-m3: A multi-view multi-modal dataset for 3d human pose estimation in outdoor scenes. arXiv preprint arXiv:2308.00628 (2023) [8](#), [9](#), [10](#)
15. Fan, B., Zheng, W., Feng, J., Zhou, J.: Lidar-hmr: 3d human mesh recovery from lidar. IEEE Transactions on Multimedia (2025) [2](#), [4](#), [9](#), [10](#)

16. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021) [4](#)
17. Geman, D., Geman, S.: Bayesian image analysis. In: Disordered systems and biological organization, pp. 301–319. Springer (1986) [6](#)
18. Guan, S., Xu, J., He, M.Z., Wang, Y., Ni, B., Yang, X.: Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(4), 5070–5086 (2022) [9](#)
19. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots* **34**(3), 189–206 (2013) [2](#), [8](#)
20. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022) [13](#)
21. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9223–9232 (2023) [1](#), [3](#)
22. Jiang, H., Cai, J., Zheng, J.: Skeleton-aware 3d human shape reconstruction from point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5431–5441 (2019) [10](#)
23. Jiang, P., Osteen, P., Wigness, M., Saripalli, S.: Rellis-3d dataset: Data, benchmarks and analysis. In: 2021 IEEE international conference on robotics and automation (ICRA). pp. 1110–1116. IEEE (2021) [3](#)
24. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7122–7131 (2018) [4](#)
25. Kocabas, M., Athanasiou, N., Black, M.J.: Vibe: Video inference for human body pose and shape estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5253–5263 (2020) [9](#)
26. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2252–2261 (2019) [4](#), [7](#), [9](#)
27. Li, J., Zhang, J., Wang, Z., Shen, S., Wen, C., Ma, Y., Xu, L., Yu, J., Wang, C.: Lidarcap: Long-range marker-less 3d human motion capture with lidar point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20502–20512 (2022) [2](#), [4](#), [10](#)
28. Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3383–3393 (2021) [9](#)
29. Li, Y., Li, S., Liu, X., Gong, M., Li, K., Chen, N., Wang, Z., Li, Z., Jiang, T., Yu, F., et al.: Sscbench: A large-scale 3d semantic scene completion benchmark for autonomous driving. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 13333–13340. IEEE (2024) [1](#), [3](#)
30. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9087–9098 (2023) [1](#), [3](#), [11](#), [14](#)
31. Li, Z., Liu, J., Zhang, Z., Xu, S., Yan, Y.: Cliff: Carrying location information in full frames into human pose and shape estimation. In: European Conference on Computer Vision. pp. 590–606. Springer (2022) [4](#), [5](#), [9](#), [10](#)

32. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1954–1963 (2021) [4](#)
33. Lin, K., Wang, L., Liu, Z.: Mesh graphormer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12939–12948 (2021) [4](#)
34. Liu, G., Rong, Y., Sheng, L.: VoteHmr: Occlusion-aware voting network for robust 3d human mesh recovery from partial point clouds. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 955–964 (2021) [10](#)
35. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866. Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3596711.3596800> [4](#), [7](#)
36. Miao, R., Liu, W., Chen, M., Gong, Z., Xu, W., Hu, C., Zhou, S.: Oc-depth: A depth-aware method for 3d semantic scene completion. arXiv preprint arXiv:2302.13540 (2023) [1](#), [3](#)
37. Moon, G., Chang, J.Y., Lee, K.M.: V2v-poseNet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In: Proceedings of the IEEE conference on computer vision and pattern Recognition. pp. 5079–5088 (2018) [10](#)
38. Moon, G., Lee, K.M.: I2l-meshNet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In: European Conference on Computer Vision. pp. 752–768. Springer (2020) [4](#)
39. Porzi, L., Bulò, S.R., Colovic, A., Kotschieder, P.: Seamless scene segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8277–8286 (2019) [14](#)
40. Shetty, K., Birkhold, A., Jaganathan, S., Strobel, N., Kowarschik, M., Maier, A., Egger, B.: Pliks: A pseudo-linear inverse kinematic solver for 3d human body estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 574–584 (2023) [9](#)
41. Shin, S., Kim, J., Halilaj, E., Black, M.J.: Wham: Reconstructing world-grounded humans with accurate 3d motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2070–2080 (2024) [9](#)
42. Sun, P., Kretschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020) [3](#)
43. Sun, Y., Bao, Q., Liu, W., Fu, Y., Black, M.J., Mei, T.: Monocular, one-stage, regression of multiple 3d people. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11179–11188 (2021) [9](#)
44. Sun, Y., Bao, Q., Liu, W., Mei, T., Black, M.J.: Trace: 5d temporal regression of avatars with dynamic cameras in 3d environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8856–8866 (2023) [9](#)
45. Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. Advances in Neural Information Processing Systems **36**, 64318–64330 (2023) [1](#), [3](#)
46. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8406–8415 (2023) [1](#), [3](#)

47. Von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European conference on computer vision (ECCV). pp. 601–617 (2018) [8](#), [9](#)
48. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17850–17859 (2023) [1](#), [3](#)
49. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21729–21740 (2023) [1](#), [3](#)
50. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation. Advances in neural information processing systems **35**, 38571–38584 (2022) [4](#)
51. Yang, Z., Wang, S., Manivasagam, S., Huang, Z., Ma, W.C., Yan, X., Yumer, E., Urtasun, R.: S3: Neural shape, skeleton, and skinning fields for 3d human modeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13284–13293 (2021) [2](#), [4](#)
52. Yu, Z., Shu, C., Deng, J., Lu, K., Liu, Z., Yu, J., Yang, D., Li, H., Chen, Y.: Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. arXiv preprint arXiv:2311.12058 (2023) [4](#), [11](#), [13](#)
53. Yu, Z., Shu, C., Sun, Q., Bian, Y., Wei, X., Yu, J., Liu, Z., Yang, D., Li, H., Chen, Y.: Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center. arXiv preprint arXiv:2406.10527 (2024) [4](#), [11](#), [13](#)
54. Zhai, H., Mei, J., Min, C., Chen, L., Zhao, F., Hu, Y.: Wildocc: A benchmark for off-road 3d semantic occupancy prediction. arXiv preprint arXiv:2410.15792 (2024) [3](#)
55. Zhang, A., Eranki, C., Zhang, C., Park, J.H., Hong, R., Kalyani, P., Kalyanaraman, L., Gamare, A., Bagad, A., Esteva, M., et al.: Toward robust robot 3-d perception in urban environments: The ut campus object dataset. IEEE Transactions on Robotics **40**, 3322–3340 (2024) [2](#), [4](#)
56. Zhang, J., Mao, Q., Shen, S., Wen, C., Xu, L., Wang, C.: Lidarcapv2: 3d human pose estimation with human–object interaction from lidar point clouds. Pattern Recognition **156**, 110848 (2024) [4](#)
57. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: Motionbert: A unified perspective on learning human motion representations. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 15085–15099 (2023) [9](#)