

HumanReg: Self-supervised Non-rigid Registration of Human Point Cloud

Yifan Chen Zhiyu Pan Zhicheng Zhong Wenxuan Guo Jianjiang Feng* Jie Zhou
 Department of Automation, Tsinghua University, Beijing, China
 {chenyf21, pzy20, zhongzc18, gwx22}@mails.tsinghua.edu.cn {jfeng, jzhou}@tsinghua.edu.cn

Abstract

In this paper, we present a novel registration framework, *HumanReg*, that learns a non-rigid transformation between two human point clouds end-to-end. We introduce body prior into the registration process to efficiently handle this type of point cloud. Unlike most existing supervised registration techniques that require expensive point-wise flow annotations, *HumanReg* can be trained in a self-supervised manner benefiting from a set of novel loss functions. To make our model better converge on real-world data, we also propose a pretraining strategy, and a synthetic dataset (*HumanSyn4D*) consists of dynamic, sparse human point clouds and their auto-generated ground truth annotations. Our experiments shows that *HumanReg* achieves state-of-the-art performance on CAPE-512 dataset and gains a qualitative result on another more challenging real-world dataset. Furthermore, our ablation studies demonstrate the effectiveness of our synthetic dataset and novel loss functions. Our code and synthetic dataset is available at <https://github.com/chenyifanthu/HumanReg>.

1. Introduction

Point cloud is a crucial data format in the fields of robotics and autonomous driving, where robots need to capture and analyze data from the environment dynamically. In indoor scenes, depth cameras [20], dense IR cameras [16] or a set of multi-view RGB cameras [35] are commonly used to record dynamic 3D data of the scene or objects in it. As to the outdoors, considering the need of large covering, existing 3D imaging systems [10, 12, 13, 17, 22, 23, 55, 59] often use LiDAR to achieve real-time scanning of the surroundings. This inevitably brings two problems. First, in each individual frame, the point cloud of each object is sparse due to the large distance from scanning device. In addition, such point clouds often suffer from occlusion, ambiguity, and noise. Both issues affect downstream tasks performance such as 3D reconstruction [33, 40], human pose

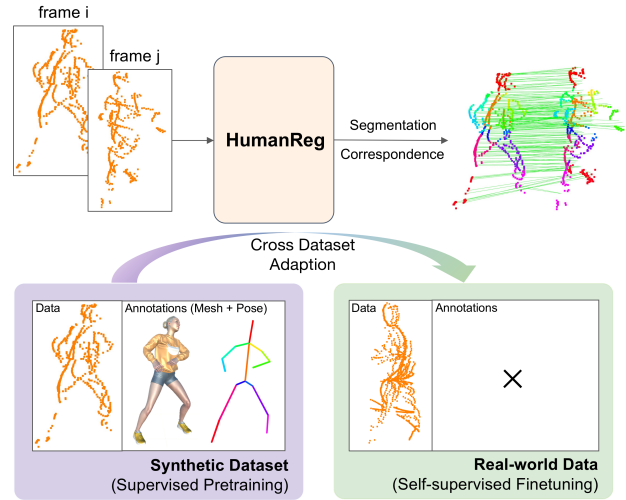


Figure 1. **HumanReg overview.** The proposed *HumanReg* framework takes a pair of human point clouds as input, simultaneously estimates the body-part segmentation for each point cloud and the scene flow between them. *HumanReg* can be pretrained on our synthetic dataset using ground-truth annotations, then adapted to unlabeled real-world data with our proposed self-supervised loss.

estimation [6, 37, 73], and object recognition [46].

A simple idea to densify dynamic object point clouds is to align the scanned frames at different times. For rigid objects like cars in the scene, the problem can be formulated as estimating the rigid transformation matrix between any two frames. Many recent works have achieved good results on this problem using traditional [7, 57, 66, 72] or learning-based methods [4, 15, 54, 65, 69]. As for deformable objects like human bodies, a feasible method is to estimate the scene flow [61], which describes the 3D motion for each point. Most of the existing learning-based methods aims at minimizing the flow loss under the ground truth supervision of scene flow [5, 25, 27, 56, 60] or point correspondences [19, 29, 38, 45]. However, obtaining scene flow annotations for real-world data is quite expensive. Although some works [34, 42] try to fit a template mesh to human point cloud and calculate the scene flow, such methods will introduce non-negligible error on sparse cases [37].

*Corresponding author

To resolve such problems, we propose HumanReg (Fig. 1), a non-rigid registration framework designed to register human point clouds captured from the same person at different times. Different from previous works that directly estimate the correspondence between two point clouds, we introduce a body-part segmentation head into our framework. This provides latent pose information for point-wise feature learning and benefits the registration process.

Registering rigid objects is usually easier than non-rigid ones because we only need to estimate a 6DoF transformation matrix. The transformation of human body is a special case, which can be regarded as a combination of several rigid parts. This is a valid assumption because each skeleton of human body is not deformable. Therefore, we can use the body-part segmentation results to guide the registration process. We formulate such rigid constraints as a novel self-supervised loss function, which can be used to train our model without the need of expensive ground-truth labels.

To give the model a better initialization and convergence on real-world data, we propose a multi-person and multi-view synthetic dataset, HumanSyn4D, to simulate human point clouds scanned by mechanical LiDARs in a large outdoor scene. Since the ground-truth correspondence and pose of each person can be generated automatically from the known template meshes, we use this dataset to pretrain both heads of our framework. Then we adapt the model to real-world data using the proposed self-supervised loss. Our experiments show that the combination of our framework, synthetic dataset, pretraining and finetuning strategy can achieve remarkable results on different types of real-world data. In summary, our contributions are:

- We propose an end-to-end human point cloud registration framework, HumanReg, that predicts the scene flow between raw human point clouds. We introduce body-part segmentation head into our framework to enhance extracted features.
- We formulate the non-rigid human registration problem as a part-rigid registration problem and design a novel self-supervised loss to train our model without the need of expensive ground-truth labels.
- We propose a multi-person synthetic dataset, HumanSyn4D, to make our model better converge on real-world data.

2. Related Work

2.1. Non-rigid Point Cloud Registration

Correspondence-based Method. Finding accurate point correspondences between point clouds is a useful solution in both rigid and non-rigid registration task. At scene level, DynamicFusion [52] finds correspondences by matching the depth map of adjacent frame based on a coarse warping field. VolumeDeform [32] computes SIFT [43] matches

of input frames to improve tracking quality. Schmidt et al. [58] also use an image-based method to extract features and reconstruct scenes. At object level, traditional methods [1, 51, 67] aim at minimizing certain type of optimization functions. Inspired by some rigid registration frameworks [3, 18, 30] using learned local or global descriptors to extract point features, 3DCODED [24] learns a global vector to transform the template into input surface. Leopard [38] enhances extracted point features with a self-attention and cross-attention module.

Scene Flow Estimation. Scene flow [61] directly describes the 3D transition between two point clouds. A few methods [5, 31] split the scene point cloud into a static background and rigid objects with different motions to obtain scene flow. FlowNet3D [41] applies a flow embedding layer and a set of upconv layers to estimate flow end-to-end. FLOT [56] and PointPWC-Net [63] use network to estimate the transportation distance and cost volume between two point clouds. PointPWC-Net also proposes a self-supervised loss to train the model. Some recent techniques [19, 28, 48] handle this problem by aligning shapes via functional map [53].

2.2. Non-rigid 3D Datasets

Unlike rigid dataset [11, 64, 70] whose point clouds are directly sampled from the surface of static object or scene, non-rigid dataset contains deforming objects and their sequences of motion.

Real-world Dataset. Most of the real-world collected datasets focus on reconstructing the surface of human body [2, 8, 26, 32, 62, 68, 71]. Their original data is mainly collected from RGB-D cameras, and they are relatively small, and the collection device has to be placed close enough to the human body to get a dense scan. In auto-driving field, KITTI [21, 22] collects a wide range of outdoor 3D scans with LiDARs fixed to a moving car. Based on KITTI, Menze et al. [49, 50] estimate 2D optical flow and project it to 3D point cloud to get sparse scene flow.

Synthetic Dataset. Although LiDAR scans can cover a large area, labeling point-wise scene flow annotations is always a challenging and error-prone task. Synthetic method has been used to solve this dilemma in recent work [9, 38, 39, 47]. Among them, FlyingThing3D [47] uses Blender to generate random 3D trajectories for everyday objects. DeformingThings4D [39] introduces a large synthetic dataset, covering a wide variety of deforming things from humanoids to animal species. As for human body, SMPL [42] uses a skinned vertex-based model to generate naked body mesh of different body shapes and poses. Ma et al. [44] proposes a framework to represent clothed human body, and uses a high-resolution body scanner to obtain dense scan sequences. However, the form of their point clouds are quite different from LiDAR data collected in a real-world large scene.

3. Method

3.1. Problem Definition

Given a pair of human point clouds $\mathbf{P} \in \mathbb{R}^{n \times 3}$ and $\mathbf{Q} \in \mathbb{R}^{m \times 3}$, where n, m are the number of points, our goal is to find a warp function $\mathcal{W} : \mathbb{R}^{n \times 3} \mapsto \mathbb{R}^{n \times 3}$ that aligns \mathbf{P} to \mathbf{Q} . In this work, we solve this problem by estimating per-point 3D flow $\mathbf{F} \in \mathbb{R}^{n \times 3}$, and the warp functions can be defined as $\mathcal{W}(\mathbf{P}) := \mathbf{P} + \mathbf{F}$.

Given a human point cloud \mathbf{P} and its corresponding ground truth pose, we can assign a label l_i for each point $\mathbf{p}_i \in \mathbf{P}$, represented the body part it belongs to

$$l_i = \arg \min_{k \in \{1, \dots, K\}} d(\mathbf{p}_i, B_k), \quad (1)$$

where B_k represents the k -th segment skeleton of the human body (from the given 3D joint locations and their topological connection method), and $d(\cdot, \cdot)$ calculate the distance from a point to line segment in \mathbb{R}^3 space. In this work, we use 15 joint points to represent body skeleton, and the detailed definition is provided in Suppl.

3.2. Architecture of HumanReg

Fig. 2 shows the overview of our proposed method.

Backbone. We utilize the ResUNet backbone [15] to extract point descriptors of input point clouds. The backbone is implemented with MinkovskiEngine [14], which defines standard neural network layers like convolutional and deconvolutional layer on 3D data, and uses sparse tensor to speedup inference and minimize memory footprint. In human registration task, we fix the size of each sparse voxel at 0.01m.

Segmentation Head. The extracted descriptors, denoted as \mathbf{D}_i , are passed through a body-part segmentation head utilized by an MLP and a softmax layer, where the predicted label for each point can be defined as

$$\hat{l}_i = \text{Softmax}(\text{MLP}(\mathbf{D}_i)). \quad (2)$$

In segmentation head, we introduce human body prior to the custom registration model. This has two advantages: 1) The body-part information can enhance the features extracted by the backbone and reduce mismatch in the correspondence head. 2) Segmentation and flow estimation will be combined to compute our self-supervised loss (Sec. 3.4).

Correspondence Head. The extracted descriptors \mathbf{D}_i are simultaneously passed through another head to estimate flow. The descriptors are first updated by an MLP layer: $\mathbf{D}_i \leftarrow \text{MLP}(\mathbf{D}_i)$. We use soft correspondence to describe the relationship between the inputs. Given the input updated descriptor $\mathbf{D}^{\mathbf{P}}, \mathbf{D}^{\mathbf{Q}}$, the soft correspondence matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$ can be computed as

$$\mathbf{C}_{ij} = -\frac{1}{t} \left\| \mathbf{D}_i^{\mathbf{P}} - \mathbf{D}_j^{\mathbf{Q}} \right\|_2, \quad (3)$$

$$\mathbf{C}_{i:} \leftarrow \text{Softmax}(\mathbf{C}_{i:}). \quad (4)$$

Here, a trainable parameter t is used to control the distance threshold in training process. We set its initial value to 0.1 and minimum value to 0.02. The softmax function is applied to each row in \mathbf{C} to make \mathbf{C} row-stochastic. We can obtain flow estimation by

$$\mathbf{F} = \mathbf{P}^w - \mathbf{P} = \mathbf{C}\mathbf{Q} - \mathbf{P}, \quad (5)$$

where \mathbf{P}^w is the warped point cloud of \mathbf{P} .

3.3. Supervised Pretraining on Synthetic Data

It's easy to acquire ground-truth labels and flow of the synthetic dataset (Sec. 4) where the template mesh of each avatar is known. Therefore, we use supervised learning to pretrain our model on synthetic data.

In the segmentation head, the human pose is used to generate ground-truth labels according to Eq. 1. Then, we use CrossEntropy criterion $\text{CE}(\cdot, \cdot)$ to calculate the supervised loss

$$\mathcal{L}_{\text{seg}} = \frac{1}{n} \sum_{i=1, \dots, n} \text{CE}(\hat{l}_i, l_i^{\text{gt}}). \quad (6)$$

In the correspondence head, we directly measure the flow loss as the Frobenius norm between ground-truth flow annotation \mathbf{F}^{gt} and the estimated flow \mathbf{F} :

$$\mathcal{L}_{\text{flow}} = \frac{1}{n} \left\| \mathbf{F}_i - \mathbf{F}_i^{\text{gt}} \right\|_F^2, \quad (7)$$

Finally, we use weighted parameters α_1, α_2 to balance the total pretrain loss

$$\mathcal{L}_{\text{pt}} = \alpha_1 \mathcal{L}_{\text{seg}} + \alpha_2 \mathcal{L}_{\text{flow}}, \quad (8)$$

3.4. Self-supervised Finetuning on Real-world Data

It is unrealistic to manually label scene flow directly on real-world collected point clouds. In this section, we propose a self-supervised objective function designed specifically for HumanReg. It consists of four parts: *Chamfer Loss*, *Smoothness Loss*, *Clustering Loss*, and *Part-Rigid Loss*.

Chamfer Loss. Chamfer loss encourages source point cloud warped close to the target.

$$\begin{aligned} \mathcal{L}_{\text{chamfer}} = & \frac{1}{n} \sum_{\mathbf{p}_i^w \in \mathbf{P}^w} \min_{\mathbf{q}_j \in \mathbf{Q}} \left\| \mathbf{p}_i^w - \mathbf{q}_j \right\|_2^2 + \\ & \frac{1}{m} \sum_{\mathbf{q}_j \in \mathbf{Q}} \min_{\mathbf{p}_i^w \in \mathbf{P}^w} \left\| \mathbf{p}_i^w - \mathbf{q}_j \right\|_2^2. \end{aligned} \quad (9)$$

Smoothness Loss. Inspired by [63], smoothness loss enforces local spatial smoothness, which means close points in space should have similar flows.

$$\mathcal{L}_{\text{smooth}} = \frac{1}{n} \sum_{\mathbf{p}_i \in \mathbf{P}} \frac{1}{|\mathcal{N}_{\mathbf{P}}(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in \mathcal{N}_{\mathbf{P}}(\mathbf{p}_i)} \left\| \mathbf{F}_i - \mathbf{F}_j \right\|_2^2, \quad (10)$$

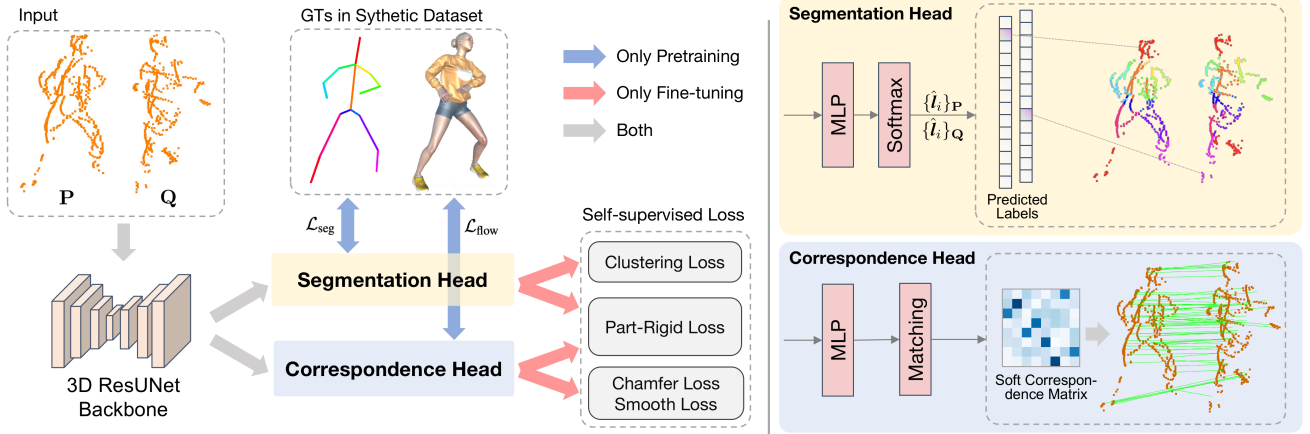


Figure 2. **Training pipeline of our proposed method.** Given the input human point clouds \mathbf{P} and \mathbf{Q} , the 3D ResUNet backbone extracts per-point features, which are then processed by a segmentation head and a correspondence head (Sec. 3.2). The two heads simultaneously output body-part segmentation of each point cloud and the soft correspondence between them. Our model is firstly pretrained on synthetic dataset with ground-truth labels and flow (Sec. 3.3). Then, a set of self-supervised loss functions (Sec. 3.4) are applied based on the estimation of both heads when finetuning on real-world data.

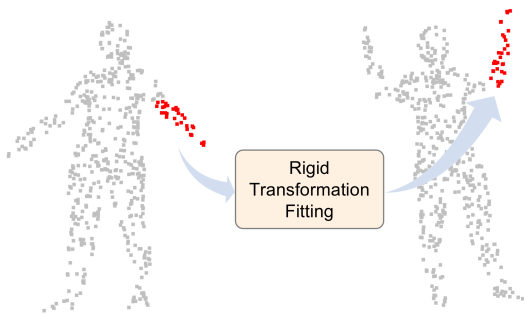


Figure 3. **Rigid fitting for body part.** We assume that the warp field of each body part is close to a rigid transformation. This assumption is used to design our part-rigid loss and refine flow during test time.

where $\mathcal{N}_{\mathbf{P}}(\mathbf{p}_i)$ is the point set of k nearest neighbors of \mathbf{p}_i .

Clustering Loss. Similar to smoothness loss, clustering loss assumes that points belonging to the same body-part should cluster together in space. It can be written as

$$\mathcal{L}_{\text{cluster}} = \frac{1}{n} \sum_{\mathbf{p}_i \in \mathbf{P}} \frac{1}{|\mathcal{N}_{\mathbf{P}}(\mathbf{p}_i)|} \sum_{\mathbf{p}_j \in \mathcal{N}_{\mathbf{P}}(\mathbf{p}_i)} \text{CE}(\hat{l}_i, \hat{l}_j). \quad (11)$$

Part-Rigid Loss. Solving the non-rigid registration problem is always more difficult than the rigid one. The rigid transformation is a 6-DoF matrix, while non-rigid warp field is composed of per-point 3D flow. However, the human body is somewhat between rigid and non-rigid. In SMPL model [42], a local part has different mesh deformations as the pose changes. But for point clouds scanned in a large scene, the slight inconsistency is ignorable compared to the sensor noise. In this way, we assume that the

warp field of each body part can be approximated by a rigid transformation.

In our model, we design a segmentation head. It not only helps to enhance extracted feature during pretraining, but also divides the original point cloud into several body parts. Thus, utilizing the output of both head in our model, we formulate the above assumption as a part-rigid loss. As shown in Fig. 3, for each body part predicted in segmentation head, we first estimate a rigid transformation for its warp field

$$\mathbf{T}_k^* = \arg \min_{\mathbf{T}_k} \|\mathbf{T}_k \circ \mathbf{P}_k - (\mathbf{P}_k + \mathbf{F}_k)\|_2, \quad (12)$$

where $\mathbf{P}_k, \mathbf{F}_k \in \mathbb{R}^{n_k \times 3}$ represents the point set of the k -th body part and the estimated flow output by correspondence head. \mathbf{T}_k^* can be decomposed into a rotation matrix $\mathbf{R}_k \in SE(3)$ and a translation vector $\mathbf{t}_k \in \mathbb{R}^3$. Our part-rigid loss describes the fitting error between the rigid transformation and the estimated scene flow of each body part

$$\mathcal{L}_{\text{rigid}} = \frac{1}{n} \sum_{k=1}^K \sum_{\mathbf{p}_i \in \mathbf{P}_k} \|(\mathbf{R}_k - \mathbf{I}) \cdot \mathbf{p}_i + \mathbf{t}_k - \mathbf{f}_i\|_2^2. \quad (13)$$

The total self-supervised loss is the weighted sum of the four type of losses

$$\mathcal{L}_{\text{total}} = \sum \beta_{\text{type}} \mathcal{L}_{\text{type}}, \quad (14)$$

where $\text{type} \in \{\text{chamfer}, \text{smooth}, \text{cluster}, \text{rigid}\}$.

During test time, we use the same assumption in Fig. 3 to refine flow estimation. After computing the rigid transformation $\mathbf{R}_k, \mathbf{t}_k$ using Eq. 12, the final flow output for each body part is

$$\tilde{\mathbf{F}}_k = \mathbf{R}_k \cdot \mathbf{P}_k + \mathbf{t}_k - \mathbf{P}_k. \quad (15)$$

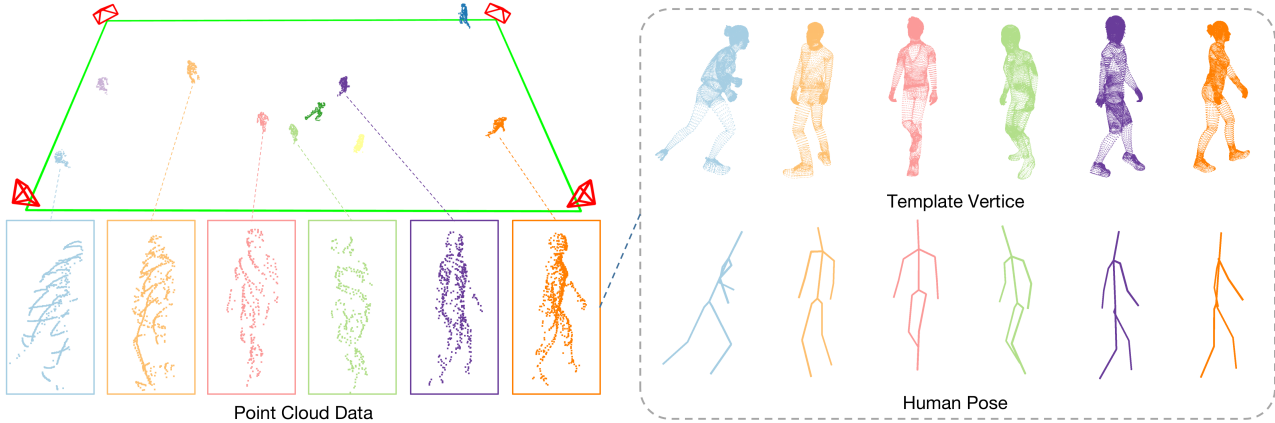


Figure 4. **A snapshot of HumanSyn4D.** Top Left: Top view of the synthetic scene, \triangle represents simulated LiDAR and green lines are the boundaries of the field. Bottom Left: Scanned human point clouds in one frame. Top Right: Ground-truth template mesh vertices. Bottom Right: Ground-truth human pose of each person.

4. HumanSyn4D

Training our network in a supervised manner requires a sufficient amount of human data with pose and flow ground truth at point level. To provide such data, we propose HumanSyn4D, a multi-person multi-view synthetic dataset consisting of sparse point clouds, avatar’s mesh vertices in any timestamp, and 3D human pose labels. A snapshot of HumanSyn4D is shown in Fig. 4.

We develop our synthetic system on Unity platform [36] due to its flexibility and productivity. Specifically, we download ten different human 3D models from Adobe Mixamo¹ and initially place them randomly in a $30\text{m} \times 15\text{m}$ scene. We use action files to drive the deformation of the human mesh and update each person’s position in the scene.

To collect human point clouds, we place four simulated LiDARs at the four corners of the scene. The laser beam is emitted from the center of the LiDAR at a certain angle, falls on a human mesh surface and returns its distance. This acquisition method effectively simulates occlusion in the real world. We use non-repetitive sampling of the Livox Mid-40 LiDAR to emit laser beam, which can be formulated as

$$r = r_0 \cos(\omega t + \theta_0), \quad (16)$$

where ω is the angular velocity of the wedge rotation in LiDAR, r_0 is the maximum scanning radius in pixel and θ_0 is a random initial angle. This equation is defined in polar coordinates on a virtual imaging plane. We eventually convert it to Cartesian coordinates and merge the points of each person.

¹<https://www.mixamo.com>

5. Experiments

5.1. Dataset and Settings

Datasets. Our experiments are conducted on two kind of human point cloud dataset, For all the datasets we keep only the points from the foreground human body.

- **CAPE-512.** CAPE dataset [44] contains 3D human body point clouds scanned with a high-resolution body scanner. Huang et al. [28] sample from its raw scans and obtain the ground-truth flow from the fitted template mesh to compose the MPC-CAPE dataset. However, each body scan in MPC-CAPE has 8192 points, which is much denser than the point cloud of outdoor scans. Therefore, we randomly sample 512 points (1/16 of the original resolution) in each scan to form the CAPE-512 dataset. We utilize the mesh template estimated from dense point clouds to obtain the ground-truth flow. CAPE-512 is used for quantitative comparison with baselines.
- **BasketballPlayer** is a much more challenging real-world dataset collected by ourselves. We use four Livox Mid-100 LiDARs to record a basketball match with ten players. Due to the fast movement and fierce confrontation of the players, there are large noises and occlusions in the data. We first calibrate the external parameters between four LiDARs, then crop the original scan to remove points from the surroundings and use a fitted plane to remove the ground. Points from different people are separated and tracked throughout the match. The density distribution of HumanSyn4D and Basketballplayer is shown in Fig. 5.

The comparison of the two datasets and our synthetic dataset is shown in Table 1.

Baselines and Training Strategy. In our work, we focus on comparing the performance of different baselines in self-supervised manner. For fairness, we pretrain all methods

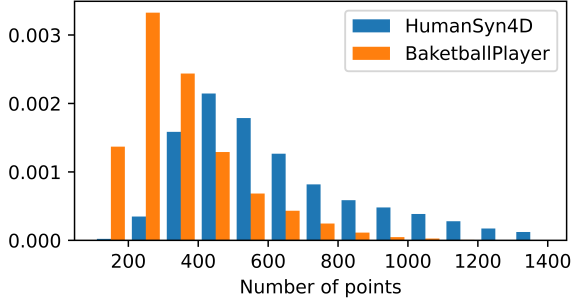


Figure 5. Histogram of numbers of points in HumanSyn4D and BasketballPlayer.

Table 1. **Comparison of Datasets.** Dataset division in experiment and differences between three datasets.

Dataset	# Train / Val / Test	Label	Real	Large Scene
HumanSyn4D	15862 / 4532 / 2266	✓	-	✓
CAPE-512	12060 / 3192 / 836	✓	✓	-
BasketballPlayer	13916 / 4000 / 2000	-	✓	✓

on HumanSyn4D dataset and compare their unsupervised training performance on CAPE-512 and BasketballPlayer. We compare the following baselines and formulate different training strategies for them.

- **Non-learned:** Coherent Point Drift method (CPD) [51]. We directly test it on target datasets.
- **Supervised Only:** FLOT [56]. We train it on HumanSyn4D and then test it on target datasets directly.
- **Both Supervised and Self-supervised:** PointPWC-Net (PointPWC) [63] where we modify the number of points in the feature pyramid to accommodate sparse input, Synorim [28] where we use the full pipeline, and our method (HumanReg). We pretrain these frameworks on HumanSyn4D in a supervised manner first, then finetune them on target datasets with their self-supervised loss.

Evaluation Metrics. We use following metrics to evaluate flow quality: (1) 3D End-Point Error (EPE3D): flow error $\|\mathbf{F}^{\text{gt}} - \hat{\mathbf{F}}\|_2$ over all points where $\hat{\mathbf{F}}$ donates the predicted flow. (2) 3D Accuracy Strict (AccS): the percentage of points with $\text{EPE3D} < 0.05\text{m}$. We removed the relative error part in [25] because we think this will underestimate the error when the human displacement is large. (3) 3D Accuracy Relax (AccR): the percentage of points with $\text{EPE3D} < 0.1\text{m}$. (4) Outlier Ratio: the percentage of points with $\text{EPE3D} > 0.2\text{m}$.

Parameters Setting. Following [15, 28], we use a 4-layer U-Net as our backbone, and the output feature dimension is 64. k is set to 5 when searching for neighbors of a point in Eq. 10 and 11. In supervised learning, we balance segmen-

tation and flow loss with weights of $\alpha_1 = 0.1, \alpha_2 = 0.9$. In self-supervised learning, we balance Chamfer / Smoothness / Clustering / Part-Rigid losses with the weights 1.0 / 1.0 / 0.1 / 10.0, respectively. We divide CAPE-512 and BasketballPlayer into non-consecutive sequences of length 4 at equal intervals. In the experiment, we register frames 1-3 with frame 4 of each sequence and calculate metrics.

5.2. Quantitative Results on CAPE-512

Since the ground-truth flow is provided in CAPE-512, we can use it to quantitatively compare the performance of different methods. As shown in Table 2, our method achieves state-of-the-art performance in EPE3D, accuracy and outlier ratio. Even without test refinement, it has 28.2% lower error compared to the nearest baseline. Our method also exhibits a much stronger ability to reduce outliers. Notably, our method obtains a great boost with the help of test refinement, making EPE3D, AccS, AccR and outlier ratio better by 29.1%, 28.3%, 1.8% and 9.8%, respectively. This proves that our part-rigid assumption is valid for registering human point clouds.

Referring to the visualization results in Fig. 6, HumanReg can best align the human point clouds, especially for extremities and large moving parts. This is because our method introduces body part information, which can jointly optimize the points of a certain part. The comparison results on sparse point clouds demonstrate the effectiveness of our ideas, and it also shows that our method can successfully densify point clouds by aligning adjacent frames.

Above results show our method can achieve remarkable performance under the setting of 512 points. Although this can simulate most real-world scenarios, we continue to reduce the number of points to verify the robustness of our method. Specifically, we continue to sample 256 and 128 points from the original CAPE dataset. The results are shown in Table 3. Our method has comparable performance on sparser point clouds with other baselines on CAPE-512. It’s worth noting that ALL other self-supervised baselines failed to converge when the number of points dropped below 512. This proves that our method is more robust to sparse points inputs.

5.3. Qualitative Results on BasketballPlayer

BasketballPlayer is a much more challenging dataset with multiple people moving quickly in a large scene. The performance on it reflects the alignment ability of the human point clouds collected in real outdoor scenes. Due to the lack of ground truth, we only made a qualitative comparison as shown in Fig. 7. The results show that our method can more accurately densify the human point clouds and ensure the correctness of human shape.

Table 2. **Quantitative comparison on CAPE-512.** We report the mean and standard deviation metrics of all sequences. \uparrow / \downarrow means higher / lower is better. *w/o refine* is the result without refinement step during test time. The best numbers are highlighted in **boldface**.

Method	Supervised Pretraining	Self-supervised Fine-tuning	EPE3D \downarrow (cm)	AccS \uparrow (%)	AccR \uparrow (%)	Outlier \downarrow (%)	Time \downarrow (s)
CPD [51]	-	-	9.44 \pm 2.90	19.3 \pm 10.8	68.5 \pm 19.3	4.80 \pm 7.18	2.28
FLOT [56]	flow	-	7.36 \pm 4.87	59.0 \pm 24.1	81.5 \pm 21.0	7.72 \pm 12.07	0.16
PointPWC [63]	flow	\checkmark	6.32 \pm 4.02	67.0 \pm 20.2	85.0 \pm 15.7	7.11 \pm 10.53	0.13
Synorim [28]	flow	\checkmark	6.62 \pm 3.32	55.2 \pm 17.9	84.8 \pm 14.1	3.88 \pm 6.29	0.63
Ours (w/o refine)	flow + joint	\checkmark	4.54 \pm 1.55	66.7 \pm 10.1	95.7 \pm 6.2	0.51 \pm 3.36	0.28
Ours	flow + joint	\checkmark	3.22 \pm 1.73	85.6 \pm 11.4	97.4 \pm 6.3	0.46 \pm 3.65	0.32

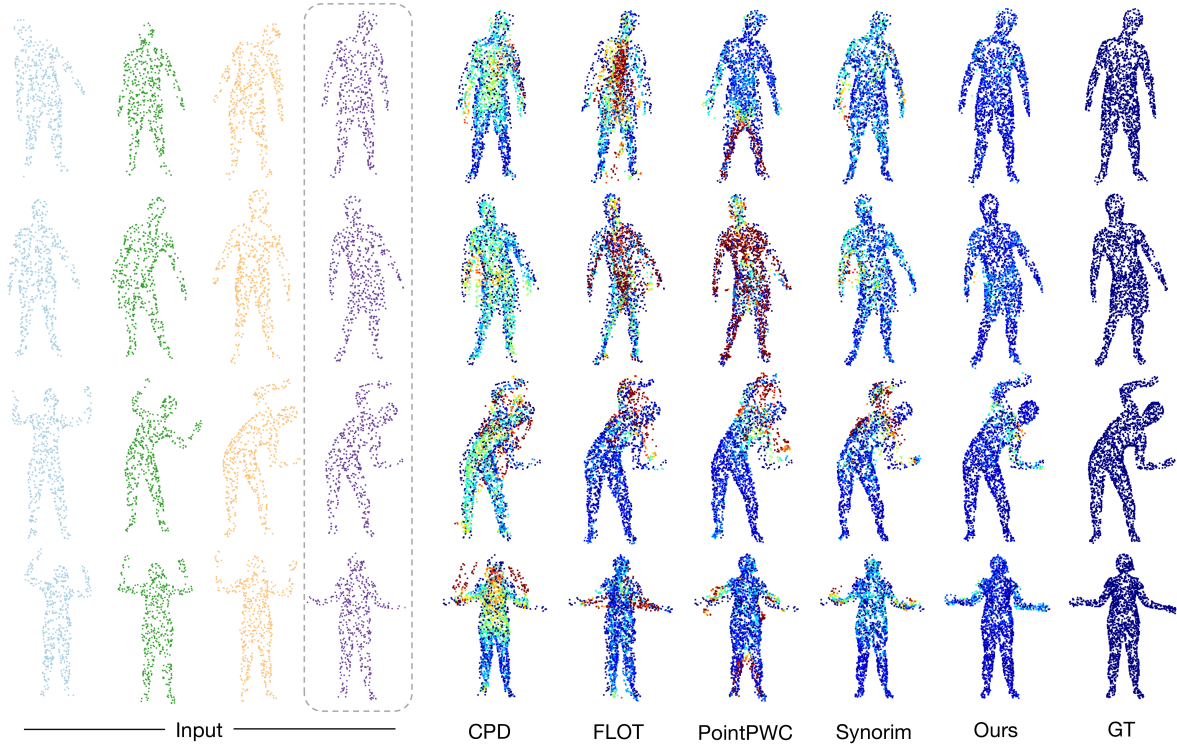



Figure 6. **Visual comparison results on CAPE-512.** Left: Input 4 frames of sparse point clouds distinguished by 4 different colors. The dashed rectangle is the reference frame to which other frames are aligned. Right: Registered point clouds and per-point L2 error are shown with colormap (0cm  20cm).

Table 3. **Results on sparser point clouds of CAPE dataset.** Besides CAPE-512, we continue to sample 256 and 128 points from the original CAPE dataset and report the registration results.

# Points	EPE3D	AccS	AccR	Outlier
512	3.22	85.6	97.4	0.46
256	6.19	47.0	88.4	1.59
128	8.25	28.5	73.1	3.11

6. Ablation Studies

We conducted five sets of experiments on CAPE-512, with the settings and results are shown in Table 4. Among them, No.1 and No.2 use the data of CAPE-512 to train from scratch without pretraining process. No.3 uses pretrained model and tests it on CAPE-512 without further finetuning. No.4 and No.5 follow the complete process from pretraining to finetuning.

Ablation on HumanSyn4D Dataset. Comparing the result of No.2 and No.5, the pretraining process reduces the EPE3D by 53.4% and improves the strict accuracy rate from

Table 4. **Comparison of different training strategies and learning loss of HumanReg.** We show the results of five sets of experiments with different training strategies and losses on CAPE-512 as ablation studies.

No.	Pretrain on HumanSyn4D	Self-supervised Learning Loss				Test Refinement	Test Metrics			
		Chamfer	Smooth	Cluster	Non-rigid		EPE3D	AccS	AccR	Outlier
1	-	✓	✓	-	-	-	8.69	40.1	74.9	8.08
2	-	✓	✓	✓	✓	✓	6.91	57.4	81.5	5.50
3	✓	-	-	-	-	✓	7.68	61.5	78.9	10.83
4	✓	✓	✓	-	-	-	5.03	58.6	94.2	0.57
5	✓	✓	✓	✓	✓	✓	3.22	85.6	97.4	0.46

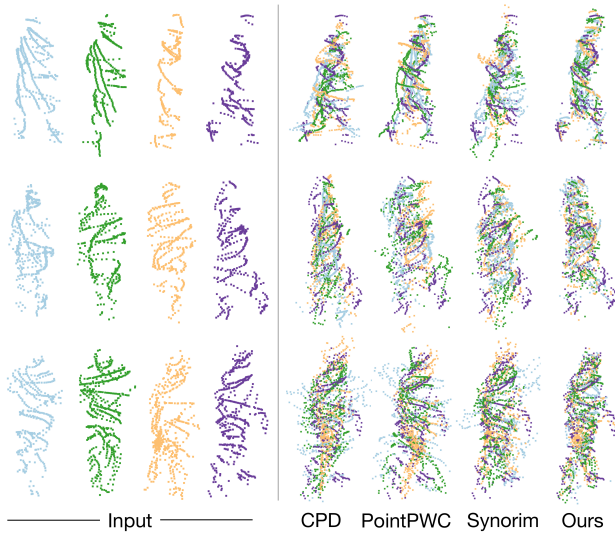


Figure 7. **Qualitative comparison results on BasketballPlayer.** Different colors are used to distinguish different frames.

57.4% to 85.6%. This is mainly because learning body-part segmentation directly on unlabeled data is difficult. Since we add a sufficient number of human poses to our synthetic dataset, pretraining can make the model have a good initial value and converge on new datasets.

Ablation on Finetuning Process. It is not feasible to directly transfer from the synthetic dataset to the real dataset. The No.3 experiment shows a large outlier ratio caused by the difference in data modality between the two datasets. In CAPE-512, the points are randomly sampled from the raw scan, while our synthetic dataset simulates points scanned with mechanical LiDARs. Therefore, a fine-tuning step is necessary for new data.

Ablation on Clustering and Non-rigid Loss. Comparing the result of No.4 and No.5, our proposed self-supervised loss reduces EPE3D and outlier ratio by 34.0% and 19.3%, while improving AccS and AccR by 46.1% and 3.4%. This demonstrates the importance of incorporating body part information for registering human point clouds.

7. Conclusion

In this work we propose HumanReg, a non-rigid registration method for sparse human point cloud. HumanReg combines flow estimation task with body-part segmentation, which makes correspondence matching based on point features more robust. We also introduce a novel self-supervised loss to our framework and make it possible to learn from unlabeled data. To train our model, we synthesize a labeled dataset, HumanSyn4D, and pretrain HumanReg on it. Then we finetune it on new unlabeled datasets in a self-supervised manner. The experiments show that our framework achieves state-of-the-art performance on CAPE-512 and gains satisfactory results on BasketballPlayer.

Limitations and Future Work. Despite the state-of-the-art performance, a few limitations are yet to be addressed: (1) Our method is based on pair-wise matching when aligning a sequence of point clouds, which ignores temporal information. An optimization method like [28] or temporal feature extraction module can be added in our framework. (2) Registration task on sparse point cloud is still particularly challenging. Our method still suffers from some failures when faced with real outdoor point clouds due to its sparsity and motion noise. (3) The improvement of our method to the performance of downstream tasks has yet to be proved by further experiments.

References

- [1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid ICP algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 2
- [3] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. SpinNet: Learning a general surface descriptor for 3D point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021. 2

- [4] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using PointNet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019. 1
- [5] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. PointFlowNet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019. 1, 2
- [6] Walid Bekhtaoui, Ruhan Sa, Brian Teixeira, Vivek Singh, Klaus Kirchner, Yao-jen Chang, and Ankur Kapoor. View invariant human body detection and pose estimation from multiple depth sensors. *arXiv preprint arXiv:2005.04258*, 2020. 1
- [7] Paul J Besl and Neil D McKay. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 1
- [8] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. DeepDeform: Learning non-rigid RGB-D reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7002–7012, 2020. 2
- [9] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 2
- [10] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1
- [11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2
- [12] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019. 1
- [13] Yookyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyoungwan An, and In So Kweon. KAIST multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018. 1
- [14] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 3
- [15] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523, 2020. 1, 3, 6
- [16] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 1
- [17] Peishan Cong, Xinge Zhu, Feng Qiao, Yiming Ren, Xidong Peng, Yuenan Hou, Lan Xu, Ruigang Yang, Dinesh Manocha, and Yuexin Ma. STCrowd: A multimodal dataset for pedestrian perception in crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19608–19617, 2022. 1
- [18] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3D point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018. 2
- [19] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020. 1, 2
- [20] Mingsong Dou, Henry Fuchs, and Jan-Michael Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 99–106. Ieee, 2013. 1
- [21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 2
- [22] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 2
- [23] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2D2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020. 1
- [24] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3D-CODED: 3D correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 2
- [25] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. HPLFlowNet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3254–3263, 2019. 1, 6
- [26] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015. 2

- [27] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multi-BodySync: Multi-body segmentation and motion estimation via 3D scan synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7108–7118, 2021. 1
- [28] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 5, 6, 7, 8
- [29] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, pages 265–278. Wiley Online Library, 2020. 1
- [30] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3D point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021. 2
- [31] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3D scene analysis by point cloud accumulation. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 674–690. Springer, 2022. 2
- [32] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14*, pages 362–379. Springer, 2016. 2
- [33] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 1
- [34] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, pages 1–1. 2014. 1
- [35] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015. 1
- [36] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018. 5
- [37] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20502–20512, 2022. 1
- [38] Yang Li and Tatsuya Harada. Leopard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5554–5564, 2022. 1, 2
- [39] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4DComplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 2
- [40] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1
- [41] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3D: Learning scene flow in 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 2
- [42] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1, 2, 4
- [43] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 2
- [44] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3D people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 2, 5
- [45] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020. 1
- [46] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 1
- [47] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 2
- [48] Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. ZoomOut: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. 2
- [49] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3D estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427, 2015. 2
- [50] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 2

- [51] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010. [2](#), [6](#), [7](#)
- [52] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015. [2](#)
- [53] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012. [2](#)
- [54] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3DRegNet: A deep neural network for 3D point registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7193–7203, 2020. [1](#)
- [55] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9552–9557. IEEE, 2019. [1](#)
- [56] Gilles Puy, Alexandre Boulch, and Renaud Marlet. FLOT: Scene flow on point clouds guided by optimal transport. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*, pages 527–544. Springer, 2020. [1](#), [2](#), [6](#), [7](#)
- [57] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009. [1](#)
- [58] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016. [2](#)
- [59] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. [1](#)
- [60] Arash K Ushani and Ryan M Eustice. Feature learning for scene flow estimation from lidar. In *Conference on Robot Learning*, pages 283–292. PMLR, 2018. [1](#)
- [61] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 722–729. IEEE, 1999. [1](#), [2](#)
- [62] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers*, pages 1–9. 2008. [2](#)
- [63] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. PointPWC-Net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3D point clouds. *arXiv preprint arXiv:1911.12408*, 2019. [2](#), [3](#), [6](#), [7](#)
- [64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. [2](#)
- [65] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. OMNet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3132–3141, 2021. [1](#)
- [66] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 38(11):2241–2254, 2015. [1](#)
- [67] Yuxin Yao, Bailin Deng, Weiwei Xu, and Juyong Zhang. Quasi-Newton solver for robust non-rigid registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7600–7609, 2020. [2](#)
- [68] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance capture of interacting characters with handheld kinects. In *ECCV (2)*, pages 828–841, 2012. [2](#)
- [69] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11824–11833, 2020. [1](#)
- [70] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3DMatch: Learning the matching of local 3D geometry in range scans. In *CVPR*, page 4, 2017. [2](#)
- [71] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. DeepHuman: 3D human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7739–7749, 2019. [2](#)
- [72] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 766–782. Springer, 2016. [1](#)
- [73] Yufan Zhou, Haiwei Dong, and Abdulmotaleb El Saddik. Learning to estimate 3D human pose from point cloud. *IEEE Sensors Journal*, 20(20):12334–12342, 2020. [1](#)