CIMI4D: A Large Multimodal Climbing Motion Dataset under Human-scene Interactions —Supplementary Material

Ming Yan[1,2,3*] Xin Wang[1,3*] Yudi Dai[1,3] Siqi Shen[1,3†] Chenglu Wen[1,3]
Lan Xu[4] Yuexin Ma[4] Cheng Wang[1,3]
[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University
[2]National Institute for Data Science in Health and Medicine, Xiamen University
[3]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, School of Informatics, Xiamen University
[4]Shanghai Engineering Research Center of Intelligent Vision and Imaging, ShanghaiTech University

Thanks for reading this document. In this supplementary material paper, we will describe the detail of the CIMI4D dataset in Appendix A, the cross dataset evaluation for CIMI4D in Appendix B, and the experimental setup and more results in Appendix C.

A. CIMI4D dataset

A.1. Coordinates

Coordinate Systems. We define three coordinate systems: 1) IMU coordinate system $\{I\}$: origin is at the pelvis joint of the first SMPL model, and $X/Y/Z$ axis is pointing to the right/upward/forward of the human. 2) LiDAR Coordinate system $\{L\}$: origin is at the center of the LiDAR, and $X/Y/Z$ axis is pointing to the right/forward/upward of the LiDAR. 3) Global/World coordinate system $\{W\}$: the scene's coordinate we manually define. We use the right subscript $k, k \in Z^+$ to indicate the index of a frame, and the right superscript, I or L or W (default to W), to indicate the coordinate system that the data belongs to. For example, the 3D point cloud frames from LiDAR is represented as $P^L = \{P_k^L, k \in Z^+\}$

Coarse calibration. Before data capturing, the actor stands facing or parallel to a large real-world object with a flat face, such as a wall or a square column. His right/front/up is regarded as the scene's $X/Y/Z$ axis direction, and the midpoint of his ankles' projection on the ground is set as the origin. After the data are collected, we manually find the first frame's ground plane and the object's plane, and then calculate their normal vector $g = [g_1, g_2, g_3]^\top$ and $m = [m_1, m_2, m_3]^\top$, respectively. The coarse calibration matrix R_{WL} from the LiDAR starting position to the world coordinate $\{W\}$ is calculated as:

$$R_{WL} = \begin{bmatrix} e_1 & e_2 & e_3 & 0 \\ m_1 & m_2 & m_3 & 0.2 \\ g_1 & g_2 & g_3 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $[e_1, e_2, e_3]^\top = m \times g$ and h is the height of the LiDAR from the ground. Based on the definition of IMU coordinate system $\{I\}$, the coarse calibration matrix R_{WI} from $\{I\}$ to $\{W\}$ is defined as: $R_{WI} = \big[(1,1,-1)(2,3,1)(3,2,1)(4,4,1)\big]_{triad}$

A.2. Notation

We use the right subscript $k, k \in Z^+$ to indicate the index of a frame, and the right superscript, I or L or W (default to W), to indicate the coordinate system that the data belongs to. For example, the 3D point cloud frames from LiDAR is represented as $P^L = \{P_k^L, k \in Z^+\}$ and the 3D scene is represented as $\boldsymbol{S}$. M_k^W indicates the k-th frame in human motion $M = (T, \theta, \beta)$ in world coordinate system, where T is the $N \times 3$ translation parameter, θ is the $N \times 24 \times 3$ pose parameter, and β is the $N \times 10$ shape parameter. N represents the number of input temporal point cloud frames. We use the Skinned Multi-Person Linear (SMPL) [8] body model Φ to map k-th frame's motion representation M_k to its triangle mesh model, $V_k, F_k = \Phi(M_k)$, where body vertices $V_k \in \mathbb{R}^{6890\times3}$ and faces $F_k \in \mathbb{R}^{13690\times3}$.

A.3. Reconstruction Scene

In the research of human-scene interaction, we consider that accurate scene reconstruction is vital for the method understanding. Previous works reconstruct scenes using depth

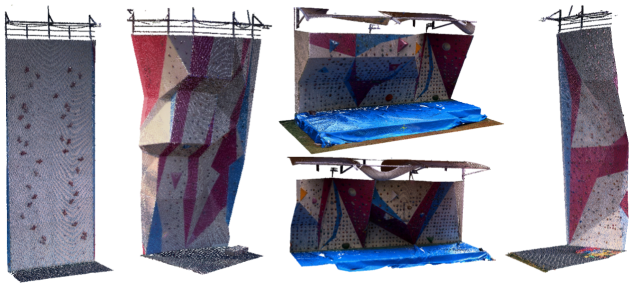

Figure 1. CIMI4D provides high-quality 3D reconstruction of RGB point cloud scenes.

cameras [3, 4, 12, 13] with much lower accuracy than LiDAR and cannot check large scenes. CIMI4D uses Trimble X7 to scan 3D scene information and rebuild the precisely measured scene in space. We provide 7 high-precision reconstruction scenes with a total point cloud amount of 40M, as shown in Fig. 1.

A.4. Blending Optimization Loss

We utilize scene and physical constraints to perform a blending optimization of pose and translation to obtain accurate and scene-natural human motion $\boldsymbol{M^W}$ annotation. The following constraints are used: the limb contact constraint $\mathcal{L}_{lc}$ encourages reasonable hand and foot contact with the scene mesh without penetrating. The limb sliding constraint $\mathcal{L}_{ls}$ eliminates the unreasonable slippage of the limbs during climbing. The smoothness constraint $\mathcal{L}_{smooth}$ makes the translation, orientation, and joints remain temporal continuity. The mesh to point constraints $\mathcal{L}_{sp}$ minimizing the distance between constructed SMPL vertices to the point clouds of human body. Please refer to the supplementary material for detailed formulation of the constraints.

The optimization is expressed as:

$$\begin{aligned} \mathcal{L} &= \lambda_{lc}\mathcal{L}_{lc} + \lambda_{ls}\mathcal{L}_{ls} + \mathcal{L}_{smooth} + \lambda_{sp}\mathcal{L}_{sp} \\ \boldsymbol{M} &= \arg\min_{\boldsymbol{M}} \mathcal{L}(\boldsymbol{M}|T^W, \theta^I, R^W, \boldsymbol{S}) \end{aligned} \tag{2}$$

where λ_{lc}, λ_{ls}, λ_{smooth}, λ_{sp} are coefficients of loss terms. $\mathcal{L}$ is minimized with a gradient descent algorithm that optimize $\boldsymbol{M^W} = (T, \theta)$. $\boldsymbol{M^W}$ is initialized according to Paper Sec 3.3.

Limb contact Loss. This loss is defined as the distance from a stable foot or hand to its nearest neighbor in the scene vertices. First, we detect the foot and hand state based on its movements. The movement is calculated based on the set of vertices of hands and feet. One limb is marked as stable if its movement is smaller than $3cm$ and smaller than another limb (foot or hand)'s movement. We obtain the contact environment near the stable limb through a neighbor search. The limb contact loss is $\mathcal{L}_{lc} = \mathcal{L}_{\text{lc}_{feet}} + \mathcal{L}_{\text{lc}_{hand}}$.

$$\mathcal{L}_{\text{lc}_{feet}} = \frac{1}{l}\sum_{j=1}^{l-1}\sum_{v\in VF^{\mathcal{SF}_j}} \frac{1}{|VF^{\mathcal{SF}_j}|}\|v_f - \widetilde{v_f}\cdot pf_j\|_2 \tag{3}$$

$$\mathcal{L}_{\text{lc}_{hand}} = \frac{1}{l}\sum_{i=1}^{l-1}\sum_{v\in VH^{\mathcal{SH}_i}} \frac{1}{|VH^{\mathcal{SH}_i}|}\|v_h - \widetilde{v_h}\cdot ph_i\|_2 \tag{4}$$

where $\widetilde{v_f}$ and $\widetilde{v_h}$ is homogeneous coordinate of v_f and v_h. $VF^{\mathcal{SF}_j}$ and $VH^{\mathcal{SH}_i}$ are the sets of the vertices of a stable foot $\mathcal{SF}_j$ and a stable hand $\mathcal{SH}_i$. The loss is average over all frames of a sequence with length l. l represents a subset of N during parallel training.

Limb sliding Loss. This loss reduces the motion's sliding on the contact surfaces, making the motion more natural and smooth. The sliding loss is defined as the distance of a stable limb over every two successive frames: $\mathcal{L}_{ls} = \mathcal{L}_{\text{ls}_{feet}} + \mathcal{L}_{\text{ls}_{hands}}$.

$$\mathcal{L}_{\text{ls}_{feet}} = \frac{1}{l}\sum_{j=1}^{l-1}\|\mathbb{E}(VF^{\mathcal{SF}_{j+1}}) - \mathbb{E}(VF^{\mathcal{SF}_j})\|_2 \tag{5}$$

$$\mathcal{L}_{\text{ls}_{hands}} = \frac{1}{l}\sum_{i=1}^{l-1}\|\mathbb{E}(VH^{\mathcal{SH}_{i+1}}) - \mathbb{E}(VH^{\mathcal{SH}_i})\|_2 \tag{6}$$

where $\mathbb{E}(\cdot)$ calculates the center of the vertices list.

Smooth Loss. The smooth loss includes the translation term $\mathcal{L}_{trans}$ and the joints term $\mathcal{L}_{joints}$.

$$\mathcal{L}_{smooth} = \lambda_{trans}\mathcal{L}_{trans} + \lambda_{joints}\mathcal{L}_{joints} \tag{7}$$

The $\mathcal{L}_{trans}$ smooths the trajectory T of human (the translation of the pelvis) through minimizing the difference between LiDAR and a human's translation difference. The smooth term is as follows:

$$\mathcal{L}_{trans} = \frac{1}{l}\sum_{j=1}^{l-1}\max(0, \|T^L_{j+1} - T^L_j\|_2 - \|T_{j+1} - T_j\|_2) \tag{8}$$

where T^L_j is the translation of LiDAR at k-th frame, and T_k is the translation we optimized for. The $\mathcal{L}_{joints}$ is the term that smooths the motion of body joints in global 3D space, which minimizes the mean acceleration of the joints. For this loss, we only consider stable joints on the torso and the neck. Let $\delta^s_j = J^s_j - J^s_{j-1}$ represent the difference of joints

Metrix	Test \ Train	LiDARHuman26M	CIMI4D	LidarHuman26M+CIMI4D
MPJPE	LiDARHuman26M	79.31	264.04	82.40
	CIMI4D	358.13	115.93	107.77
PMPJPE	LiDARHuman26M	66.72	137.10	69.87
	CIMI4D	222.11	86.38	80.61
PVE	LiDARHuman26M	101.64	340.70	105.86
	CIMI4D	422.65	136.83	126.83
ACCEL	LiDARHuman26M	4.52	7.95	4.42
	CIMI4D	12.39	2.59	2.81
PCK0.5	LiDARHuman26M	0.95	0.61	0.94
	CIMI4D	0.50	0.90	0.92

Table 1. Cross-dataset evaluation for the 3D human pose estimation task.

between consecutive frame. $\mathcal{L}_{joints}$ is defined as follows.

$$\mathcal{L}_{joints} = \frac{1}{l}\sum_{j=1}^{l-1} \|\delta_{j+1}^{s} - \delta_{j}^{s}\|_2 \tag{9}$$

Since the static scenes are collected in Paper Sec 3.1, we design a method to segment human point clouds as annotation data. For each frame of dynamic LiDAR output, we manually register to the same coordinate system of the IMU to obtain the RT matrix. Next, the human body in the multi-frame dynamic scene is manually removed to generate a sparse static scene. For each frame of point cloud, the points within the threshold range of the sparse scene are eliminated to obtain the segmented human point cloud $\mathcal{P}_i$. For each segmented human point cloud $\mathcal{P}_i$.

SMPL to point loss. For each estimated human meshes, we use Hidden Points Removal (HPR) [5] to remove the invisible SMPL vertices from the perspective of LiDAR. Then, we use Iterative closest point (ICP) [10] to register the visible vertices to $\mathcal{P}$, which is segmented human point clouds. We re-project the human body SMPL in the LiDAR coordinate to select the visible human body vertices V'. For each frame, We use $\mathcal{L}_{sp}$ to minimize the 3D Chamfer distance between human points $\mathcal{P}_i$ and vertices V'_i. For each frame, the $\mathcal{L}_{stp}$ constraint is regularized with the following equation:

$$\begin{aligned}\mathcal{L}_{sp} = &\frac{1}{|\mathcal{P}|}\sum_{p_i\in\mathcal{P}} \min_{v_i\in V'} \|p_i - v_i\|_2^2 + \\ &\frac{1}{|V'|}\sum_{v_i\in V'} \min_{p_i\in\mathcal{P}} \|v_i - p_i\|_2^2\end{aligned} \tag{10}$$

B. Cross-dataset Evaluation

In this section, we evaluate the quality of CIMI4D through cross-dataset evaluation. We have shown that the LiDAR point cloud based approaches perform better in the pose estimation tasks in the submitted paper. We use LiDARCap as the baseline method to evaluate the quality of CIMI4D.

Evaluation metrics In this section and in Appendix C, we report Procrustes-Aligned Mean Per Joint Position Error (PMPJPE), Mean Per Joint Position Error (MPJPE), Percentage of Correct Keypoints (PCK), Per Vertex Error (PVE), and Acceleration error(m/s^2) (ACCEL). Except ACCEL and PCK, error metrics are measured in millimeters.

LiDARCap is trained based on the training sets of LiDARHuman26M [7], CIMI4D, and the two combined. And then, it is evaluated based on the testing sets of LiDARHuman26M [7], CIMI4D. The results are depicted in Tab. 1.

As it is shown in Tab. 1, when LiDARCap was trained on one dataset, it performed poorly on the other dataset. Especially, when it is trained on LiDARHuman26M and is tested on CIMI4D, its performance is worse than the opposite way. This indicates that the daily actives contained in the LiDARHuman26M are not sufficient for MoCap algorithms to learn climbing actions in-depth. Further, as it is shown in the table, if LiDARCap was trained based on the combination of the two datasets, its performance on the two dataset is significantly increased. This indicates that *CIMI4D is a necessary addition to the current human motion datasets.*

For the PMPJPE metric, after being trained on both datasets, the error of LiDARCap on LiDARHuman26M increases compared to the model trained only on LiDARHuman26M. This suggests LiDARHuman26M complements CIMI4D better than the opposite. Similar conclusions can be drawn from the results of other metrics that CIMI4D is more challenging with more diverse data, which yields stronger generalizability than the LiDARHuman26M dataset.

C. Tasks and Benchmarks

C.1. Pose Estimation Benchmark

The models used to evaluate the pose estimation tasks are listed as follows.

1. LiDARCap [7], which estimates human pose based on LiDAR point clouds. It is trained based on the LiDARHuman26M dataset.

2. LiDARCap$\star$, which is trained based on the CIMI4D dataset.

3. P4Transformer$\star$ [1], which uses a 4D convolution transformer networks for spatio-temporal modeling in point cloud sequences. The model is trained using the CIMI4D dataset.

4. VIBE [6], a RGB video-based algorithm. It is trained based on its original dataset.

5. VIBE$\diamond$, it is fine-tuned based on the CIMI4D dataset.

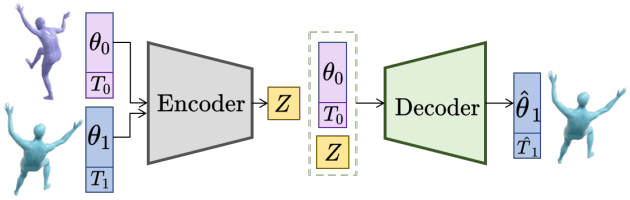

Figure 2. **Motion Prediction CVAE Architecture**. The inputs of encoder are the previous pose θ_0 with translation T_0 and the current pose θ_1 with translation T_1 during training. The decoder reconstructs $\hat{\theta}_1$ with translation $\hat{T}_1$ by sampling the encoder distribution Z.

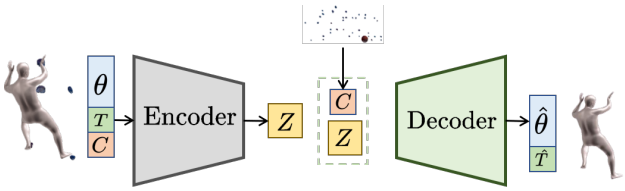

Figure 3. **Motion Generation CVAE Architecture**. The inputs of encoder are the pose θ with translation T and the rock points C during training. The decoder reconstructs $\hat{\theta}$ with translation $\hat{T}$ by sampling the encoder distribution Z.

6. MAED [11], an RGB based algorithm which is trained based on its original dataset.

7. MAED$\diamond$, fine-tuned based on the CIMI4D datasets.

8. DynaBOA [2], an RGB based algorithm which is trained based on its original dataset.

9. PROX [4], a scene-aware pose estimation algorithm based on RGB images.

10. PROX$\diamond$, it is fined tune based on the CIMI4D dataset. PROX relies on the 2D-joints provided by OpenPose, which performs poorly on CIMI4D. We improve PROX through using the ground-truth orientations of root hip joint as input.

11. LEMO [13], a scene-aware pose estimation algorithm based on RGB images.

Training Details. For LiDARCap [7], we retrain the model on CIMI4D dataset using the PyTorch framework for 200 epochs with Adam optimizer and the batch size is set to be 6. The learning rate and decay rate are set to be 1×10^{-4}. One NVIDIA GeForce RTX 3090 Graphics Card is utilized for training. P4Transformer [1] is retrained on CIMI4D dataset, and the hyperparameters are set to be the same as LiDARCap. For the other algorithms, we use their default settings when training or fine-tuning.

Pose Estimation Models based on LiDAR and scene are significantly better than models based on RGB, illustrating the importance of 3D information for Human-scene understanding.

C.2. Motion Prediction and Generation

The motion prediction task aims to predict the pose and the translation of a person in the future frames based on history frames and current frames. To evaluate the performance of the motion prediction task, we devise a simple baseline. Its architecture is depicted in Fig. 2, our motion prediction baseline used a Conditional Variational Autoencoder (CVAE) motivated by HuMoR [9]. The encoder consists of 5-layer MLPs with ReLU activation function, while the decoder are 4-layer MLPs with ReLU activation. We take the KL divergence to regularize the distribution of the encoder output to be near the Gaussian distribution. The learning rate, optimizer, and batch size are the same as the pose estimation tasks. And we experiment on original HuMOR too.

For the motion generation task, the simple architecture of the baseline is demonstrated in Fig. 3. In general, its neural network is the same as motion prediction except that the inputs and outputs are different. The learning rate and batch size are the same as the motion prediction task.

References

[1] Hehe Fan, Yi Yang, and Mohan S. Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14199–14208, 2021. 3, 4

[2] Shanyan Guan, Jingwei Xu, Michelle Z He, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 4

[3] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 2

[4] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2282–2292. IEEE, 2019. 2, 4

[5] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, pages 24–es. 2007. 3

[6] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and

shape estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5252–5262, 2020. 3

[7] 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. 3, 4

[8] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, Oct. 2015. 1

[9] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11488–11499, October 2021. 4

[10] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009. 3

[11] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13033–13042, 2021. 4

[12] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 180–200, Cham, 2022. Springer Nature Switzerland. 2

[13] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 11323–11333. IEEE, 2021. 2, 4