Supplementary Material: *mRI*: Multi-modal 3D Human Pose Estimation Dataset using mmWave, RGB-D, and Inertial Sensors

Sizhe An
University of Wisconsin-Madison
sizhe.an@wisc.edu

Yin Li
University of Wisconsin-Madison
yin.li@wisc.edu

Umit Ogras
University of Wisconsin-Madison
uogras@wisc.edu

A Supplementary Material

This document complements the main paper by describing: (1) results of pose estimation using additional metrics related to rehabilitation (A.1); (2) an analysis of our 3D pose refinement used to obtain ground-truth pose for our dataset (A.2); (3) details of our implementation and benchmark (A.3); (4) details of mmWave imaging (A.4); and (5) visualization of our pose estimation results (A.5). Paper checklist is attached as the final part of the supplement.

For sections, figures, tables, and equations, we use numbers (e.g., Table 1) to refer to the main paper and capital letters (e.g., Table A) to refer to this supplement.

A.1 Further Analysis of Pose Estimation Results

We report additional evaluation metric, the mean average error (MAE) of joints angle to supplement our main results in the paper (using MPJPE and PA-MPJPE). The metric is widely considered to evaluate rehabilitation-specific movements — a main focus of our dataset. We only consider **Protocol 2** here since it has all rehabilitation-related movements.

Joints angle. We use the joint coordinates estimated by our models to find the angles between critical joints. *mRI* focuses on the four commonly used joint angles: left & right elbow angles and left & right knee angles. The elbow angle is found using the shoulder, elbow, and wrist positions. First, we obtain the bone length between the shoulder and elbow and the length between the elbow and wrist using joint coordinates. Then, the angle is calculated using triangulation from the law of cosines. Similarly, the knee angles are obtained using the hip, knee, and ankle positions. The ground truth angle is computed using the refined ground truth 3D coordinates, and we calculated MAE between the ground truth and each modality. Table A shows detailed results of joins angle MAE. We observe that under **S1**, RGB modality yields below 10° for all joints, while mmWave and IMUs lead to larger errors regarding the elbow angles (>10°). This behavior is observed since the movement of the upper limbs is larger than that of the lower limbs for most movements. The setting of **S2** yields higher errors than under **S1** since **S2** requires the model to generalize to unseen subjects, which is arguably more challenging.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

Modality	Setting	Left elbow	Left knee	Right elbow	Right knee
mmWave	S1	18.7±0.2	2.9±0.1	16.0±0.2	3.2±0.1
	S2	24.5±2.3	10.4±1.3	22.9±2.9	11.6±1.3
RGB	S1	9.0±0.1	8.3±0.1	**9.3**±0.1	7.7±0.1
	S2	11.5±0.6	14.8±1.6	11.1±0.7	14.0±1.5
IMUs	S1	**7.9**±0.1	**2.6**±0.1	11.3±0.2	**2.4**±0.1
	S2	8.4±1.0	5.6±0.2	9.7±0.8	5.7±0.2

Table A: MAE of joints angle (°) for mmWave, RGB, and IMUs. We report the mean and standard deviation of MAE averaged across multiple splits under both our settings (**S1** & **S2**).

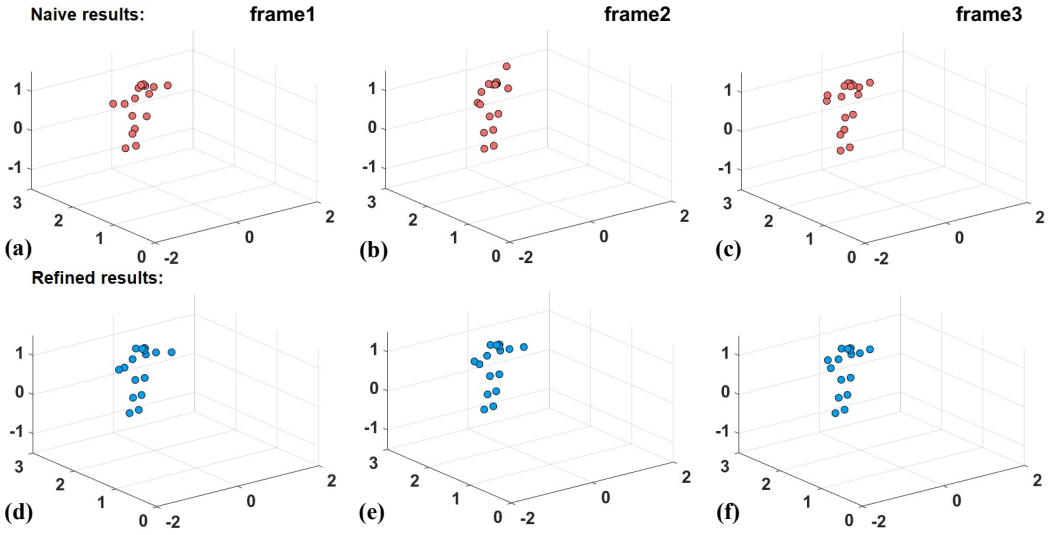

Figure A: An example of comparisons between poor naive and refined 3D pose ground truth. The subject is bending the right arm in three continuous frames from left to right. The first row shows naive results and the second row shows refined results.

A.2 Analysis of 3D Joints Refinement and Quality

We provide further analysis of our 3D pose refinement used to obtain ground-truth poses for the dataset. There are three terms co-optimized in the objective function:

$$\min_{\{\mathbf{p}_i\}} \sum_{i=1}^{\mathbb{Z}} \left(\left\| P^l \mathbf{p}_i - \mathbf{q}_i^l \right\| + \| P^r \mathbf{p}_i - \mathbf{q}_i^r \| \right) + \sum_{j}^{bonelist} \| \mathbf{B}_j - median(\mathbf{B}) \| + \sum_{i=1}^{\mathbb{Z}-1} \left\| \mathbf{p}_{i+1} - \mathbf{p}_i \right\|, \quad \text{(A)}$$

. The first term represents the reprojection errors of the two cameras. The second term enforces equal bone length across all frames in the same video (i.e., the same subject). Finally, the third term imposes temporal smoothness of the 3D joint coordinates. Figure A shows an example of naive 3D pose with *poor quality* (obtained from direct triangulation), and the refined 3D pose after our optimization. We can observe that the refined pose is more stable as only the right arm moves while the lower body parts hardly move, which is the case in reality. Overall, the average objective decreases from 176 to 83, more than 50% for all subjects.

To validate the reliability of the obtained 3D joints, we further annotate a subset of the whole dataset and calculate the error. Specifically, we manually annotate 2D keypoints of the images, randomly sampled from subjects and movements. Then, we obtain the re-project 2D keypoints using refined 3D joints and camera parameters via camera calibration. Finally, we calculate the mean absolute percentage error (MAPE), and the percentage of correct keypoints threshold at 50% of the head segment length (PCKh) between the 2D keypoints from the model and the re-projection. The MAPE is 1.5%, and PCKh is 98.92. These quantitative results show that the proposed method of obtaining 3D joints is reasonably accurate. Figure B compares manual annotated 2D keypoints and re-projected 2D keypoints from refined 3D joints. Blue dots represent manual annotations, and red dots show the re-projection keypoints. We can observe that keypoints from the two methods almost overlap.

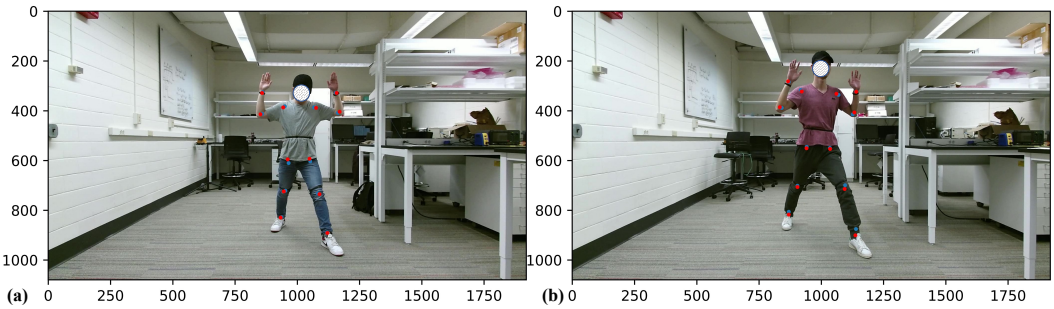

Figure B: A comparison of manual annotated 2D keypoints and re-projected 2D keypoints from refined 3D joints. Blue dots represent manual annotations and red dots show the re-projection keypoints.

A.3 Benchmark and Implementation Details

We now describe implementation details of methods considered in our benchmark. We use PyTorch [6] to implement all our models. Intel Xeon Gold 6242R @ 80x 4.1GHz and NVIDIA GeForce RTX 3090 are used to train these models. The code and pre-trained models will be open-sourced to facilitate the research area [1]. Both raw data and synchronized data are released to the public as well. All material published is made available under the following Creative Commons license: Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Data-split. For **Setting 1 (S1 Random Split)**, we set three different random seeds to split the data to 80% and 20% as training and testing set, respectively. For **Setting 2 (S2 Split by Subjects)**, we selected subset of the subjects to split the data, generated by three random seeds as well. Three different splits we used in the paper are shown as follows: (1) $[17, 13, 11, 15]$, (2) $[9, 7, 20, 8]$, and (3) $[3, 16, 7, 2]$. For example, $[17, 13, 11, 15]$ means that subject 17, 13, 11, 15 are used for testing and the rest for training.

mmWave-HPE. We follow [1] for the implementation for mmWave-HPE model. The input layer of the CNN takes the stacked 5-channel feature tensors. Two consecutive convolution layers follow the input layer with 16 and 32 channels, respectively. After the convolutions, the output is fed to the first fully-connected (FC) layer with 512 neurons. The final output of CNN contains 51 neurons, representing 3D coordinates for the 17 joints. All activation functions are Relu except for the final FC layer, where we use linear activation. Dropout layers are used after the convolution and fully connected layers to avoid excessive dependency on specific neurons. The model converges within around 50 epochs with early stopping settings.

RGB-HPE. We employ HRNet-W32 [8] (with bounding boxes from Mask RCNN [4]) to detect 2D keypoints of human body parts in all RGB frames from both cameras. W32 in HRNet represents the width of the high-resolution nets in the last three stages. The pre-trained model from [5] is utilized for 3D joints estimation. It "lift" 2D keypoints from a sequence of frames into 3D joints.

IMU-HPE. We follow [11] for IMU calibration, normalization, and features generation. Each IMU has its own coordinate system. As a result, two steps are needed to make the output compatible with neural network models. First, *calibration:* transforming the raw inertial measurements into the same reference frame. Second, *normalization:* transforming the leaf joint inertia into the root's space and scaling it to a suitable size for the network input. This method calculates the transition matrices for each sensor before capturing the movements, and it requires subjects to perform a 'T pose' before the experiments. The feature tensors extracted and transformed by this method capture the joint rotation and acceleration effectively such that multilayer perceptron (MLP) or CNN can regress the 3D joints with these features. We use a similar model as mmWave-HPE, except the input tensors are only 1-channel feature tensors for IMUs. The model converges within around 30 epochs with early stopping settings.

Skeleton-based action detection. We re-purpose an existing model [12] for the skeleton-based action detection. Specifically, the model takes a sequence of estimated 3D poses from individual

[1] https://sizhean.github.io/mri

Symbol	Description	Values	Symbol	Description	Values
f_c	Starting frequency	77 GHz	θ_{res}	Angle resolution	9.55°
T_c	Chirp signal duration	32 μs	N_{RX}	No. of RX antennas	4
B	Bandwidth	3.20 GHz	N_P	Maximum points detectable per frame	64
S	Slope of chirp signal	100 MHz/μs	N_{TX}	No. of TX antennas	3
N	No. of chirps per frame	96	v_{res}	Velocity resolution	0.35 m/s
d_{res}	Range resolution	4.69 cm	v_{max}	Maximum Velocity	5.69 m/s

Table B: List of major parameters and variables related to mmWave and their values for mmWave point cloud generation.

modality as inputs. These poses are further encoded into a feature pyramid using a multi-scale transformer. Shared classification and regression heads check the feature pyramid, thus producing an action candidate at every timestamp.

A.4 mmWave Imaging

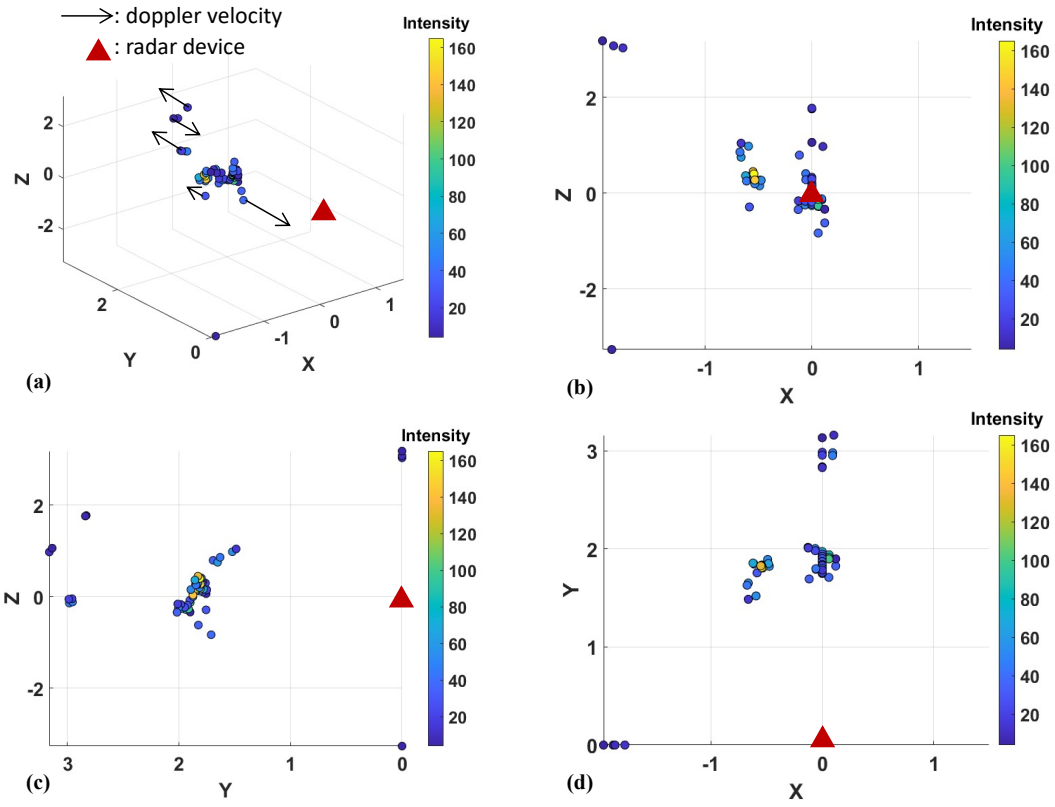

Figure C: mmWave point cloud representation for one frame. (a), (b), (c), and (d) shows the 3D view, front view, side view, and top view, respectively.

We follow [1] for the mmWave point cloud generation including software and hardware setup, data pre-process, and follow [2] for fusing the continuous frames point cloud to reduce the effect of sparsity. For the comprehensive details and math derivation of mmWave imaging background, please refer to [7, 9, 10, 3]. Figure C shows a sample input frame from different views. The red marker represents the radar location. Figure C(a) shows that point positions in 3D view, while the other plots show the front view, side view, and top view. Specifically, Figure C(a) illustrates the Doppler velocity, which indicates the relative velocity from the detected point to the radar. The colors in the figures

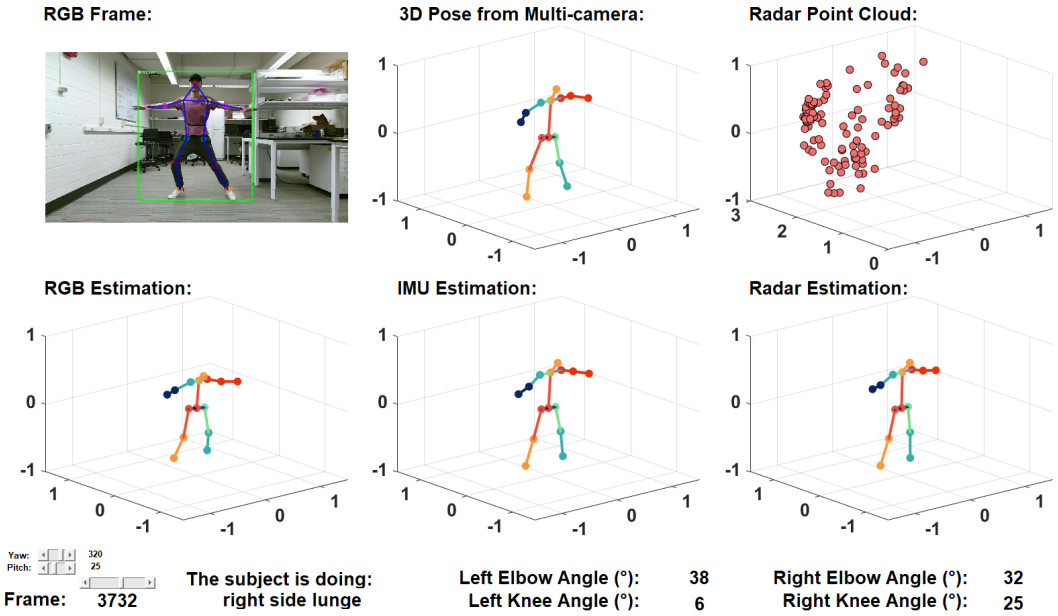

Figure D: Visualization of body poses from one subject performing right side lunge. The units in axes are meter.

represent the energy intensity of the reflected signals. Table B lists the key parameters we used to generate the mmWave point cloud.

A.5 Additional Visualization

Figure D shows one subject performing right side lunge. Figure E and F demonstrate pose estimation results from different camera pose. The results are displayed with the RGB frame from the camera, the refined 3D pose, and the 3D point cloud from mmWave radar. The first row, from left to right: RGB frame with detected human bounding box and 2D keypoints, the refined 3D pose from multiple cameras, and mmWave radar point cloud signal. The second row, from left to right: estimated 3D pose from a single RGB camera, IMU signals, and mmWave radar point cloud. The captions include the action label and four commonly used joint angles: left & right elbow angles and left & right knee angles.

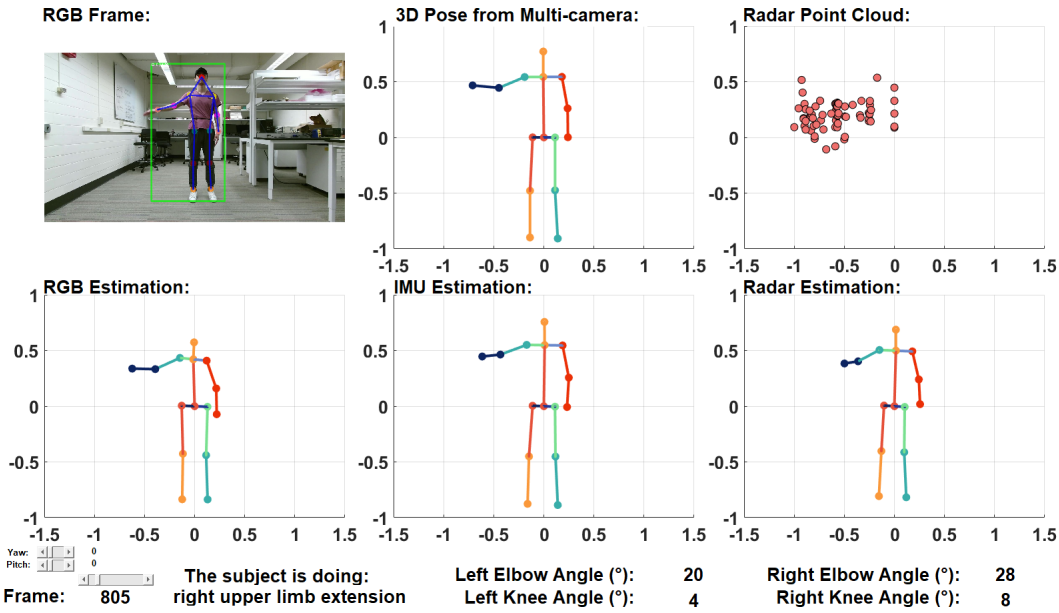

Figure E: Dataset visualization when $yaw = 0°, pitch = 0°$. The units in axes are meter.

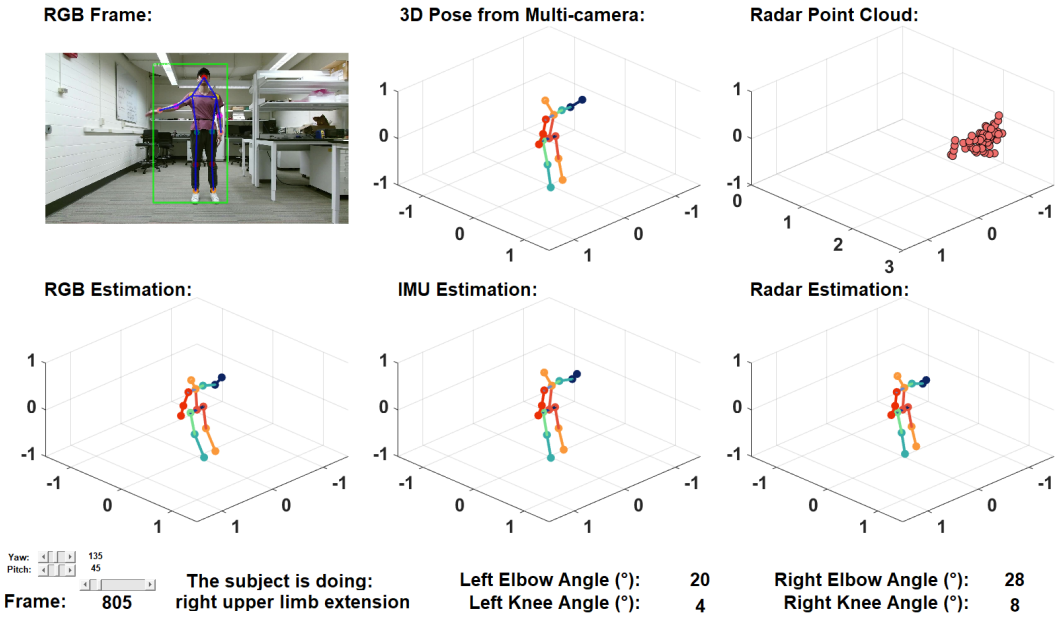

Figure F: Dataset visualization when $yaw = 135°, pitch = 45°$. The units in axes are meter.

Paper checklist

For all authors:

- Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
- Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
- Did you discuss any potential negative societal impacts of your work? [Yes] , please check Section 5 (**Ethic statements**).

- Did you describe the limitations of your work? [Yes] , as a dataset paper, all our algorithms and models reported in the paper are just baseline.

If you ran experiments:

- Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] , please check our project page.
- Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] , in Section 4 and supplementary materials.
- Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] , in Section 4 and supplementary material.
- Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] , in supplementary materials.

If you are using existing assets (e.g., code, data, models) or curating/releasing new assets:

- If your work uses existing assets, did you cite the creators? [Yes]
- Did you mention the license of the assets? [Yes] All material published is made available under the following Creative Commons license: Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). We mention this in the supplement.
- Did you include any new assets either in the supplemental material or as a URL? [Yes]
- Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] , we used others' algorithm and pre-trained models.
- Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] , please check Section 5.

If you used crowdsourcing or conducted research with human subjects:

- Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] , in **Participant recruitment and consent** of Section 3 and Section 5.
- Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] , we partially describe it in **Participant recruitment and consent** of Section 3 and Section 5. The full IRB approvals will be released after we confirm with school.
- Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] , those are specified in IRB approvals. Each subject gets 20 dollars Amazon gift card after completing the experiments. In total we spent 400 dollars incentives.

References

[1] S. An and U. Y. Ogras. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s):1–22, 2021.

[2] S. An and U. Y. Ogras. Fast and scalable human pose estimation using mmwave point cloud. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, page 889–894, 2022.

[3] V. Dham. Programming chirp parameters in ti radar devices. *Application Report SWRA553, Texas Instruments*, 2017.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *in Proc. of IEEE Intl. Conf. on Computer Vision*, pages 2961–2969, 2017.

[5] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Pytorch. Pytorch Mobile. https://pytorch.org/mobile/home/ accessed 8 Jul. 2021, 2022.

[7] S. Rao. Introduction to mmwave sensing: Fmcw radars. *Texas Instruments (TI) mmWave Training Series*, 2017.

[8] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[9] Texas Instruments. mmWavetutorial. https://www.ti.com/lit/pdf/swra553 accessed 29 Sep. 2020, 2014.

[10] Texas Instruments. mmWavefundamentals. https://www.ti.com/lit/spyy005 accessed 8 Apr. 2021, 2020.

[11] X. Yi, Y. Zhou, and F. Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40(4), 08 2021.

[12] C. Zhang, J. Wu, and Y. Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, 2022.