Demo: Real-time mmWave Radar Human Sensing Testbed

Ruofeng Liu[1], Shuai Wang[2], Shuai Wang[2], Wenchao Jiang[3], Weiwei Chen[4], Ruili Shi[2], Luoyu Mei[2], Taiwei Lin[2]

[1]Michigan State University [2]Southeast University [3]Singapore University of Technology and Design [4]Shanghai University

liuruofe@msu.edu,{shuaiwang_iot,shuaiwang}@seu.edu.cn,wenchao_jiang@sutd.edu.sg
chen.ava.0012@gmail.com,{shiruili,lymei-,220222061}@seu.edu.cn

Abstract

Millimeter-wave (mmWave) radar is emerging as a promising sensor for various human sensing tasks. Deep learning is frequently applied in radar-based applications, which typically require extensive data collection and labeling. In this demo, we present a low-cost hardware setup and a cross-platform software pipeline that automatically captures radar data of human activities, labels ground truth, and tests inference models in real time. The effectiveness of the testbed is demonstrated through real-time human pose estimation.

ACM Reference Format:

Ruofeng Liu, Shuai Wang, Shuai Wang, Wenchao Jiang, Weiwei Chen, Ruili Shi, Luoyu Mei, Taiwei Lin. 2024. Demo: Real-time mmWave Radar Human Sensing Testbed. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3636534.3698860

1 INTRODUCTION

MmWave Radar is an emerging sensor for various human sensing tasks (e.g., pose estimation, activity recognition, and fall detection). By estimating the range, velocity, and angle of arrival of targets, it generates 3D point clouds. However, due to hardware limitations and the inherent characteristics of mmWave signals, such as specular reflection, these point clouds tend to be sparse and noisy. To address these challenges, deep learning methods are often employed in mmWave sensing applications [5–8], which involve extensive data collection, ground truth labeling, model training, and testing. In this work, we introduce our mmWave radar

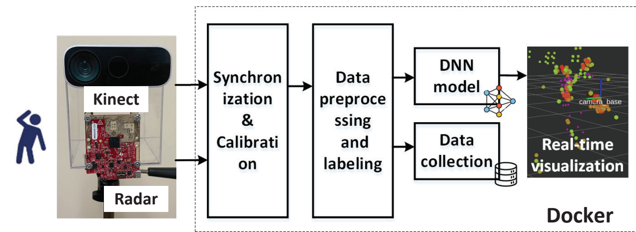

Figure 1: The data processing pipeline for real-time mmWave Radar human sensing testbed.

human sensing testbed and data processing toolchain, designed to streamline the development of deep learning-based radar applications. The testbed comprises an mmWave radar sensor and a co-located RGB-D camera, which automatically labels ground truth data (e.g., body skeleton). Careful calibration of the radar and Kinect ensures precise alignment of the sensors in both space and time. Additionally, our testbed supports real-time deployment of inference models, enabling developers to compare predictions with ground truth data on the fly. The key features of our testbed include:

1. **Real-time**. The testbed integrates radar point cloud collection, preprocessing, ground truth labeling (including body skeleton and SMPL human mesh), and real-time model inferences into the Robot Operating System (ROS). The collected data are visualized in Rviz, enabling real-time comparison.
2. **Cross-platform**. Integrating various software tools is a complex task. For instance, the ROS driver for mmWave radar is only compatible with Ubuntu 18.04. To resolve such compatibility issues, we developed a Docker image that encapsulates the entire software toolchain. This Docker image enables users to install all necessary tools with a single click, eliminating dependency issues and simplifying the setup process.
3. **Low-cost**. The testbed consists of various low-cost devices including a TI IWR6843ISK-ODS ($175) [1], a Azure Kinect RGB camera($399)[2], a corner reflector ($50), and a tripod($50). The software toolchain can be deployed on a commercial PC with an enthusiast-class graphics card (e.g., NVIDIA GeForce RTX 3090).

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11
https://doi.org/10.1145/3636534.3698860

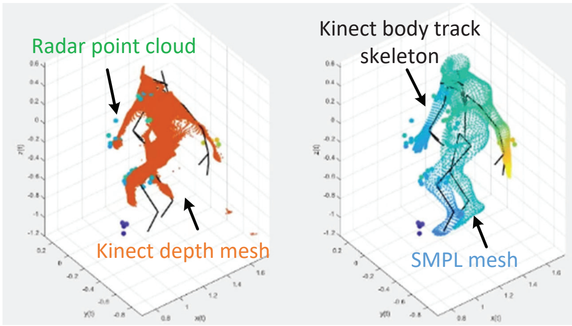

Figure 2: Preprocessed Radar and Kinect data.

2 OVERVIEW

Fig1 shows the overall data processing pipeline of the developed toolchain consisting of the following modules:

Synchronization and calibration. Radar and Kinect operate at different frame rates and have slight differences in viewpoint, calling for synchronization in time and space. Temporally, each radar frame is matched with the closest Kinect frame based on timestamps. For spatial calibration, a 55mm corner reflector is placed at various positions within a 3×3×3 grid, with a 1-foot distance between adjacent locations. Radar point clouds and colorized depth images of the corner reflector are collected and used to compute a 4×4 calibration matrix. This calibration procedure is performed once, and the resulting matrix is applied to preprocess human sensing data during human sensing data collection.

Data preprocessing. mmWave and Kinect data are preprocessed to obtain the deep learning model input and ground truth labels. Targeting human sensing tasks (e.g., pose estimation), the preprocessing results are mmWave radar point cloud, Kinect body skeleton, and SMPL parameters of human mesh. Specifically, the raw radar and Kinect data are first calibrated using the calibration matrix. For radar data, we use the body skeleton (produced by Azure Kinect Body Tracking [3]) to locate the target person and segment the radar point cloud belonging to the person. For the ground truth labels, we extended the tool [4] to fit the Kinect skeleton location and orientation to the skinned multi-person linear model (SMPL) and obtain the pose and shape parameters of SMPL human mesh. In addition, the preprocessing modules also handle the errors in the data (e.g., packet drop).

Data collection. The preprocessed radar data and ground truth labels are aggregated into the matrices and stored in .npy files. The files can be imported into Pytorch for neural network training and inference. Fig.2 depicts the preprocessed data in which the radar point cloud (the sparse points) and the ground truth labels (body tracking skeleton and SMPL mesh) are accurately aligned in both time and space, showing the effectiveness of our calibration method.

DNN models. The collected data can be used for training various human sensing tasks. For demonstration purposes, we use the collected data to train a model for mmWave human pose estimation. The model adopts PointNet to extract features from the radar point cloud and predict the body skeleton and SMPL mesh of the person. We integrate the trained model into the toolchain for real-time inference. The model predicts pose and publishes the results to ROS.

Real-time visualization. We utilize the ROS visualization tool, Rviz, to display raw data, ground truth labels, and model predictions. Real-time visualization allows us to quickly identify errors during data collection and qualitatively assess the model's accuracy. A demo video is available at https://youtu.be/OrWJZhMqSII.

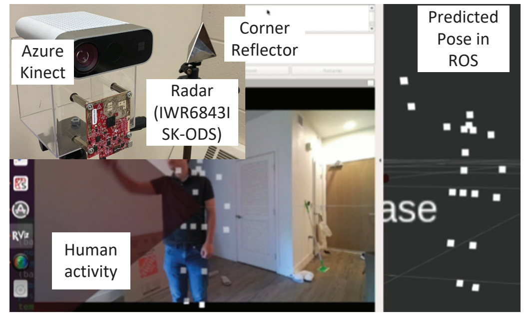

Figure 3: Hardware setup and application.

3 DEMO SETUP

Hardware. As Fig.3 shows, TI IWR6843ISK-ODS and a Azure Kinect are mounted to a tripod. A 55mm corner reflector is deployed during calibration.

Software. The software runs on Ubuntu 20.04 PC with NVIDIA GeForce GTX 1070 (driver Version: 470.57.02 and CUDA Version: 11.4). The docker engine 26.1.4 is installed. Our entire toolchain is a Ubuntu 18.04 docker image that integrates the TI mmWave Radar ROS driver, Azure Kinect ROS driver, and software components introduced in Section 2.

Scene: We demonstrate the effectiveness of the toolchain with a real-time human pose estimation demo. In the demo, a person stands in front of the hardware setup, performing various activities such as walking in place, waving, lunging, and moving back and forth. The software collects radar point clouds and uses a pre-trained model to predict the 3D human pose, represented by a skeleton, in real time. The graphical user interface (GUI) of the demo, shown in Fig. 3, displays the predicted body skeleton alongside the camera images.

Requirement: We would like to get a portion of room ($5m \times 5m$ or a corner of a room) to set up sensors and screen. A person will perform various activities in this area.

4 CONCLUSION

We demonstrate a low-cost and cross-platform mmWave radar human sensing testbed. In the future, we plan to integrate other sensors (e.g., IMU and optitrack) into the testbed.

References

[1] Iwr6843isk-ods. https://www.ti.com.cn/tool/cn/IWR6843ISK-ODS, May 2023.

[2] Sensor capture + azure kinect + refinement workflow. https://www.depthkit.tv/tutorials/azure-kinect-microsoft-volumetric-capture-depth-workflow-depthkit, May 2023.

[3] Azure kinect body tracking sdk 1.1.2. https://www.microsoft.com/en-us/download/details.aspx?id=104221, Sep 2024.

[4] Fitting smpl parameters by 3d-pose key-points. https://github.com/Dou-Yiming/Pose_to_SMPL, Sep 2024.

[5] Hao Kong, Xiangyu Xu, Jiadi Yu, Qilin Chen, Chenguang Ma, Yingying Chen, Yi-Chao Chen, and Linghe Kong. m3track: mmwave-based multi-user 3d posture tracking. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 491–503, 2022.

[6] Shuai Wang, Dongjiang Cao, Ruofeng Liu, Wenchao Jiang, Tianshun Yao, and Chris Xiaoxuan Lu. Human parsing with joint learning for dynamic mmwave radar point cloud. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(1):1–22, 2023.

[7] Hongfei Xue, Qiming Cao, Yan Ju, Haochen Hu, Haoyu Wang, Aidong Zhang, and Lu Su. M4esh: mmwave-based 3d human mesh construction for multiple subjects. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 391–406, 2022.

[8] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. mmmesh: Towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 269–282, 2021.