RF-Based 3D Skeletons

Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh,
Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, Antonio Torralba
Massachusetts Institute of Technology

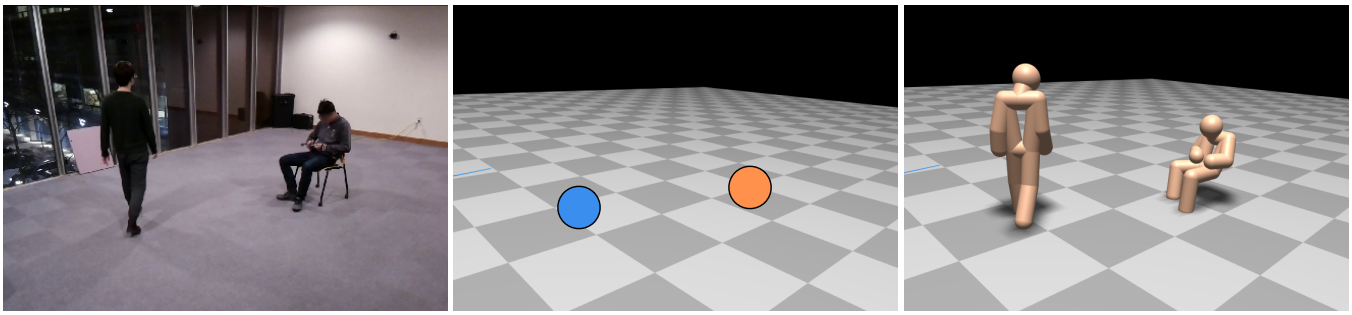


Figure 1: Left: RGB image. Middle: RF-based localization results. Right: 3D skeletons from our system.

ABSTRACT

This paper introduces RF-Pose3D, the first system that infers 3D human skeletons from RF signals. It requires no sensors on the body, and works with multiple people and across walls and occlusions. Further, it generates dynamic skeletons that follow the people as they move, walk or sit. As such, RF-Pose3D provides a significant leap in RF-based sensing and enables new applications in gaming, healthcare, and smart homes.

RF-Pose3D is based on a novel convolutional neural network (CNN) architecture that performs high-dimensional convolutions by decomposing them into low-dimensional operations. This property allows the network to efficiently condense the spatio-temporal information in RF signals. The network first zooms in on the individuals in the scene, and crops the RF signals reflected off each person. For each individual, it localizes and tracks their body parts – head, shoulders, arms, wrists, hip, knees, and feet. Our evaluation results show that RF-Pose3D tracks each keypoint on the human body with an average error of 4.2 cm, 4.0 cm, and 4.9 cm along the X, Y, and Z axes respectively. It maintains this accuracy even in the presence of multiple people, and in new environments that it has not seen in the training set. Demo videos are available at our website: http://rfpose3d.csail.mit.edu.

SIGCOMM '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5567-4/18/08…$15.00
https://doi.org/10.1145/3230543.3230579

CCS CONCEPTS

• **Networks → Cyber-physical networks**; *Sensor networks*;
• **Computing methodologies → Machine learning**;

KEYWORDS

RF Sensing, 3D Human Pose Estimation, Machine Learning, Neural Networks, Localization, Smart Homes

ACM Reference Format:
Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, Antonio Torralba. 2018. RF-Based 3D Skeletons. In *SIGCOMM '18: ACM SIGCOMM 2018 Conference, August 20–25, 2018, Budapest, Hungary.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3230543.3230579

1 INTRODUCTION

The past decade has witnessed much progress in using RF signals to localize people and track their motion. Novel algorithms have led to accurate localization within tens of centimeters [19, 34]. Advanced sensing technologies have enabled people tracking based on the RF signals that bounce off their bodies, even when they do not carry any wireless transmitters [2, 17, 35]. Various papers have developed classifiers that use RF reflections to detect actions like falling, walking, sitting, etc. [21, 23, 32]. This literature shows that RF signals carry an impressive amount of information about people and their movements. But, how rich a description of people can one extract from the surrounding radio signals?

In this paper, we demonstrate the potential of extracting rich and detailed information about people using the radio signals that bounce off their body. Instead of simply returning a person's location, we present RF-Pose3D, a new system that can use the RF signals in the environment to extract full 3D skeletons of people including the head, arms, shoulders, hip, legs, etc. Further, the extracted skeletons are dynamic, i.e., they move and act like the original people in the scene. Fig. 1 presents the output of our system, and compares it against RF-based localization. The figure on the left shows a scene with two people. The figure in the middle illustrates the output of today's RF-based localization systems. The figure on the right shows the output of our system, which not only localizes the people, but also provides their detailed 3D skeletons and reveals their exact posture. Further, if the persons in Fig. 1(a) move, the skeletons in Fig. 1(c) would move accordingly.

Such 3D skeletons have applications in gaming where they can extend systems like Kinect to work across occlusions. They may be used by law enforcement personnel to assess a hostage scenario, leveraging the ability of RF signals to traverse walls. They also have applications in healthcare, where they can track motion disorders such as involuntary movements (i.e., dyskinesia) in Parkinson's patients.

Designing a system that maps RF signals to 3D skeletons is a highly complex task. The system must model the relationship between the observed radio waves and the human body, as well as the constraints on the location and movement of different body parts. To deal with such complexity we resort to deep neural networks. Our aim is to leverage recent success of convolutional neural network (CNN), which has demonstrated a major leap in abstracting the human pose in images and videos [6, 13, 22].

Our neural network takes as input the RF signal captured by an FMCW radio similar to the radio used in past work on localization [2]. The network operates on sliding time windows of 3 seconds. It produces a continuous 3D video of the skeletons in the scene, where for each skeleton it tracks the 3D location of 14 keypoints: head, neck, shoulders, elbows, wrists, hip, knees, and feet.

The design of RF-Pose3D is structured around three components that together provide an architecture for using deep learning for RF-sensing. Each component serves a particular function as we describe below.

(1) Sensing the 3D Skeleton: This component takes the RF signals that bounce off someone's body, and leverages deep CNN to infer the person's 3D skeleton. There is a key challenge, however, in adapting CNNs to RF data. The RF signal that we deal with is a 4 dimensional function of space and time. Thus, our CNN needs to apply 4D convolutions. But common deep learning platforms (e.g., Pytorch, Tensorflow) do not support 4D CNNs. They are targeted to images or videos, and hence support only up to 3D convolutions. More fundamentally, the computational and I/O resources required by 4D CNNs are excessive and limit scaling to complex tasks like 3D skeleton estimation.

To address this challenge, we leverage the properties of RF signals to decompose 4D convolutions into a combination of 3D convolutions performed on two planes and the time axis. We also decompose CNN training and inference to operate on those two planes. We analytically prove that our decomposition is valid and equivalent to performing 4D convolutions at each layer of the neural network. This approach not only addresses the dimensional difference between RF data and existing deep learning tools, but also reduces the complexity of the model and speed up training by orders of magnitude.

(2) Scaling to Multiple People: Most environments have multiple people. To estimate the 3D skeletons of all individuals in the scene, we need a component that separates the signals from each individual so that it can be processed independently to infer his or her skeleton. The most straightforward approach to this task would run past localization algorithms, locate each person in the scene, and zoom in on signals from that location. The drawbacks of such approach are: 1) localization errors will lead to errors in skeleton estimation, and 2) multipath effects can create fictitious people. To avoid these problems, we design this component as a deep neural network that directly learns to detect people and zoom in on them. However, instead of zooming in on people in the physical space, the network first transforms the RF signal into an abstract domain that condenses the relevant information, then separates the information pertaining to different individuals in the abstract domain. This allows the network to avoid being fooled by fictitious people that appear due to multipath, or random reflections from objects in the environment.

(3) Training: Once the network is setup, it needs training data –i.e., it needs many labeled examples where each example is a short clip (3-second) of received RF signals and a 3D video of the skeletons and their keypoints as functions of time. How do we obtain such labeled examples?

We leverage past work in computer vision which, given an image of people, identifies the pixels that correspond to their keypoints [6]. To transform such 2D skeletons to 3D skeletons, we develop a coordinated system of 12 cameras. We collect 2D skeletons from each camera, and design an optimization problem based on multi-view geometry to find the 3D location of each keypoint of each person. Of course, the cameras are used only during training to generate labeled examples. Once the network is trained, we can take the radio to new environments and use the RF signal alone to track the 3D skeletons and their movements.

RF-Pose3D has been evaluated empirically. We train and test our system using data collected in public environments around our campus.[1] The dataset has over one hundred people performing diverse indoor activities: walking, sitting, waiting for elevators, opening doors, talking to friends, etc. We train and test in different environments to ensure the network generalizes to new scenes. We summarize our results as follows:

- **Qualitative Results:** Figure 1 above provides a representative example of our results (more are provided in §8.3). The figure shows an important feature of our 3D skeletons. The radio in this experiment is situated behind the seated person, and hence captures signals from a specific perspective. Yet, RF-Pose3D generates 3D skeletons that can be shown from any perspective –e.g., you can look at them from the direction opposite to the radio.
- **Accuracy of Each Keypoint:** RF-Pose3D estimates simultaneously the 3D locations of 14 keypoints on the body. Its average error in localizing a keypoint is 6.5cm in the horizontal plane and 4.0cm along the vertical axis. To the best of our knowledge, this is the first work that localizes multiple keypoints on the human body at the same time.
- **Indoor Localization:** Once we have 3D skeletons, we can easily localize people. Our median localization error is 1.7cm, 2.8cm and 2.3cm along the X, Y and Z axes, which is a significant improvement over past work.

Contributions: This paper makes the following contributions:

- This paper is the first to extract 3D skeletons and their keypoints from RF signals. Inferring the 3D skeleton is a complex task that requires mapping 14 keypoints on the human body to their 3D locations. It also involves generalization to unseen views that are different from the view of the radio.
- The paper presents a novel CNN model that differs from all past work including models used in computer vision. The key property of this model is its ability to decompose 4D CNN to 3D convolutions over 2D planes and the time axis. This method allows us to maintain spatio-temporal relationship between human keypoints, yet operate on individual views of the signal over time, which both reduces complexity and allows for using common neural network platforms.
- The paper presents an architecture that leverages deep learning to sense humans using RF signals. Our architecture consists of a component that generates training example, a component that separates RF data from different individuals, and a sensing component that infers properties related to a particular individual. We show how to build these components using deep neural networks and multi-view optimization of visual data. We believe that this architecture as well as our camera system can be used by researchers in the field to address other RF-based sensing tasks.

[1] All experiments that involve humans satisfy our IRB requirements.

2 RELATED WORK

Related work falls in two areas.

(a) Wireless Systems: Recent years have witnessed much interest in localizing people and tracking their motion using wireless signals. The literature can be divided into: device-based and device-free systems. Device-based tracking systems localize people using the signal generated by some wireless device they carry, e.g., their cell phone [19, 34]. On the other hand, device-free wireless tracking systems do not require the tracked person to wear sensors on their body. They work by analyzing the radio signal reflected off the person's body. Different papers localize the people in the environment [2, 17], monitor their walking speed [15, 31], track their chest motion to extract breathing and heartbeats [3, 39, 41], or track the arm motion to identify a particular gesture [21, 23].

Previous papers have also tried to generate a human silhouette based on RF reflections [1, 40]. Specifically, RF-Capture [1] creates a coarse description of the human body behind a wall by collapsing multiple body parts detected at different points in time. This system is limited to a single person performing a single action, which is to walk towards the device. Further, it cannot simultaneously localize multiple keypoints on the human body. Additionally, we have introduced an earlier version of RF-Pose that addresses the case of extracting a 2D skeleton from RF signals [40]. In 2D pose estimation, the keypoints are expressed as pixels in a particular 2D view. Hence, this approach cannot generalize to different views or provide depth information. Further, the design of the CNN therein does not have a method to separate inputs from different individuals in the scene. In contrast to that work, this paper introduces a method to estimate 3D poses from RF signals. The design separates RF information from different individuals and uses a decomposed 4D convolution to track their 3D poses over time.

(b) Computer Vision: Inferring the human pose from images is a known problem in the computer vision literature. The problem comes in two flavors: 2D and 3D. 2D pose estimation has achieved remarkable success recently [6, 8, 11, 13, 16, 22, 33]. This is due to the availability of large-scale datasets of annotated 2D human poses, and the introduction of deep neural network models. In contrast, advances in 3D human pose estimation remain limited due to the difficulty and ambiguity of recovering 3D information from 2D images. Other than conventional cameras, people have also explored

the potential of estimating 3D poses with RGB-Depth cameras [37] and VICON motion capture systems [27].

Our work builds on human pose estimation in computer vision in three ways. First, we similarly use deep neural networks to address this problem. Second, we leverage a vision system called OpenPose [6] to extract 2D skeletons from images. We integrate this module in our camera system, which combines such 2D skeletons across 12 cameras to create 3D skeletons that can be used as training examples for our network. Third, our module that zooms in on the RF signal from a particular individual and separates it from the signals from other individuals is inspired by object detection in computer vision, specifically systems like R-CNN, Fast R-CNN and Faster R-CNN [9, 10, 24] which use deep neural models to generate a bounding box around objects of interest in an image (e.g., a dog).

Our work, however, is fundamentally different from all past work in computer vision. We infer 3D poses from RF signals, which is intrinsically different from extracting 3D poses from images due to basic differences between the two data types. In particular, images have high spatial resolution whereas RF signals have low spatial resolution, even when using multi-antenna systems. Second, the human body scatters visible light, but acts as a reflector for the RF bands of interest (frequencies around few GHz) [5]. Hence, only signals that fall close to the normal on the body surface are reflected back towards the radio source. As a result, at any time, only a few body parts are visible to the radio [1]. Furthermore, our neural network model differs from past work in vision, and is the first to propose 4D CNN decomposition. Even our dataset is different. Existing datasets for inferring 3D poses from images are limited to one environment or one person (e.g., Human3.6M [7]). In contrast, our dataset spans multiple environments and our scenes include multiple people. This allows our system to learn to generalize to new environments which are unseen during training.

3 PRIMER

3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) has been the main workhorse of recent breakthroughs in understanding images [14, 20], videos [29, 30] and audios [4, 38]. Below we describe the basic building blocks of a CNN that are relevant to this paper.

Deep Neural Network: A deep neural network contains multiple layers of neurons that extract information from the input signal. Each neuron receives input from the neurons in the previous layer and combines them through weights and a nonlinearity (e.g., sigmoid). Mathematically, the value of a neuron a_i^n at the n-th layer is $\sigma(\sum_j w_{ij}^n a_j^{n-1})$, where a_j^{n-1} are the neurons from the previous layer, w_{ij}^n are the weights and $\sigma(\cdot)$ is a nonlinearity.

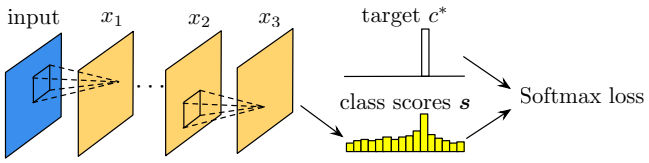


Figure 2: CNN architecture for classification task, where x_1, x_2 and x_3 are feature maps, s shows the class scores, and c^* refers to the label. The Softmax loss function compares the score vector with the label.

CNN: Different from the ordinary neural networks where each neuron is connected to all the neurons in the previous layer, each neuron in a CNN is only locally connected with a few neurons in the previous layer. Also, all the neurons in the same layer of a CNN share the weights. The value of neurons in a CNN can be computed as the convolution of a weight kernel with the neurons in the previous layer, that is $a^n = \sigma(f^n * a^{n-1})$, where $*$ is the convolution operator, and f^n refers to the weight kernel at layer n. CNNs leverage local dependencies in the data to reduce the total number of weights that the network needs to learn. Hence, they allow for much deeper networks.

Feature maps: The value of the neurons in a CNN layer are usually referred to as features maps, as they can be viewed as features that correspond to different parts of the input. As the number of layers increases, the resulting features maps capture increasingly global and more complex properties of the input signal.

Training CNN for Classification: A neural network is trained to minimize a loss function that captures the difference between the current output of the network and the desired output given some labeled examples. In classification tasks, the final layer of the network is made to output a score for each class (i.e., a vector of scores). A Softmax loss is used to measure the discrepancy between the class scores $\boldsymbol{s} = \{\boldsymbol{s}_c\}_{c=1}^K$ and the target label c^* as follow:

$$L_{\text{Softmax}}(\boldsymbol{s}, c^*) = -\log \frac{e^{\boldsymbol{s}_{c^*}}}{\sum_c e^{\boldsymbol{s}_c}}, \tag{1}$$

where $\boldsymbol{s}_{c^*}$ is the score prediction of the target class. Training a CNN means adjusting the weights of the various layers to minimize the Softmax loss. This is usually done with stochastic gradient descent and its variants [18]. Fig. 2 illustrates the basic design of a 3-layer CNN, where x_1, x_2 and x_3 are feature maps, and $\mathbf{s}$ shows the class scores. The highest score typically corresponds to the correct class indicated by the label c^*. The Softmax loss function compares the score vector with the label.

CNN inference after training: Once trained, a CNN can perform inference on new data. CNN will compute class

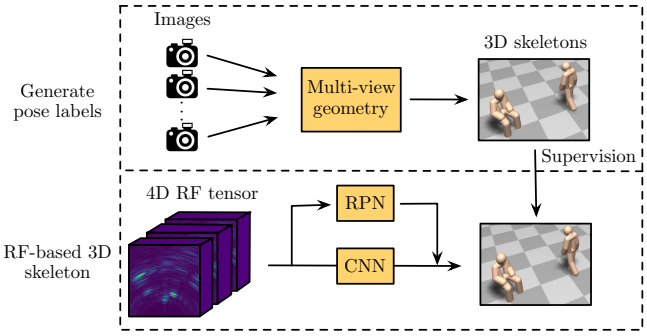


Figure 3: RF-Pose3D's system overview. Top graph shows the process of generating labeled 3D poses using our coordinated camera system. The labeled samples are used to train the model in the bottom graph. The model can be divided into two components: a region proposal network (RPN) that zooms in on RF data from one individual, and a CNN that extracts the 3D skeleton from the proposed region.

scores for the new input and predict it as the class with the highest score, that is:

$$\hat{c} = \arg\max_{c} s_c \tag{2}$$

3.2 Multi-Antenna FMCW Radio

RF-Pose3D uses a multi-antenna FMCW radio similar to the one used in [1]. The radio has a single transmit antenna, and two 1D antenna arrays for reception, one situated horizontally and the other vertically. The combination of FMCW and antenna arrays allows the radio to measure the signal from different 3D voxels in space. Specifically, the RF signals reflected from location (x, y, z) can be computed as [25]:

$$a(x, y, z, t) = \sum_{k} \sum_{i} s^{t}_{k,i} \cdot e^{j2\pi \frac{d_k(x,y,z)}{\lambda_i}}, \tag{3}$$

where $s^{t}_{k,i}$ is the i-th sample of an FMCW sweep received on the k-th receive antenna at the time index t (i.e., the FMCW index), λ_i is the wavelength of the signal at the i-th sample in the FMCW sweep, and $d_k(x, y, z)$ is the round-trip distance from the transmit antenna to the voxel at (x, y, z), and back to the k-th receive antenna.

4 OVERVIEW

RF-Pose3D is a system that estimates multi-people 3D poses based on RF signals. RF-Pose3D takes as input the RF reflections from the environment captured by a multi-antenna FMCW radio. Such reflections are a 4D function of space and time, which we refer to thereafter as a 4D RF tensor.

RF-Pose3D's design is based on a deep neural network architecture (Fig. 3). The system includes multiple components:

- A multi-camera sub-system that generates 3D poses from many 2D images taken from different viewpoints (top graph in Fig. 3). The output of this subsystem is used to provide labeled examples to train RF-Pose3D's neural networks.
- A neural network model that extracts multi-people 3D poses from RF signals (bottom graph in Fig. 3). The model is trained using labeled examples from the camera system. Once training is over, the model can infer 3D skeletons from RF signals alone. Furthermore, it can be taken to new environments that it did not see during training and would still work correctly. The model itself has two conceptual subcomponents:
  - A component that zooms in on the RF data from each individual separately. We refer to this network as the region proposal network (RPN) because it associates each person with the RF data in a particular region.
  - A component that operates on the RF data of each person and extracts his or her skeleton. We refer to this component as the CNN.

The following sections explain the above three components: the camera-system, the RPN, and the CNN. For clarity reason, we start by explaining the CNN assuming only one person in the scene. We then extend the model by adding the RPN, which takes care of separating the RF signals from different people in the scene. Finally, we explain the camera system and how it obtains labeled examples for training.

5 3D POSE ESTIMATION FROM RF

In this section, we describe our design of a CNN model that uses RF signal to estimate the 3D human pose. The problem of 3D pose estimation is defined as identifying the 3D locations of 14 anatomical keypoints on the body: head, neck, shoulders, elbows, wrists, hips, knees and ankles. We first focus on 3D pose estimation for a single person in this section, and extend it for multi-person scenarios in §6.

Manually designing a mapping from RF signals to 3D poses is an intractable task. Such a mapping has to take care of reflection properties, the presence of multi-path and other reflective objects, the deformable nature of the human body, and the constraints on the movements and locations of human body parts with respect to each other. Thus, rather than manually design filters or rules to decode 3D human poses from the RF signals, we consider neural networks, which have proved their advantage in learning complex mappings from training examples. Our goal is to design a CNN model that takes as input a 4D RF tensor (§3.2), and outputs a 3D human pose.

5.1 CNN Model

We start by formulating keypoint localization as a CNN classification problem, then design a CNN architecture that solves the problem.

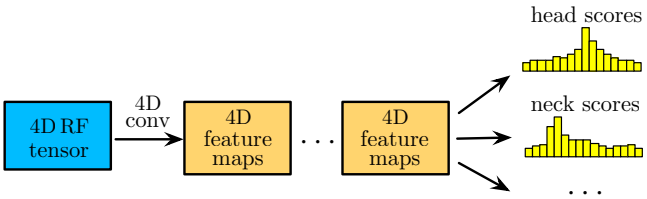


Figure 4: Illustration of RF-Pose3D's 4D CNN model. The model localizes human keypoints (e.g., head, neck, right knee) by classifying each keypoint to one voxel in space.

Keypoint localization as CNN classification: We first discretize the space of interests into 3D voxels. In our CNN classification problem, the set of classes are all 3D voxels, and our goal is to classify the location of each keypoint (head, neck, right elbow, etc.) into one of the voxels. Specifically, to localize a keypoint, our CNN outputs scores $\boldsymbol{s} = \{s_v\}_{v \in V}$ corresponding to all 3D voxels $v \in V$, and the target voxel v^* is the one that contains the keypoint. We use the Softmax loss $L_{\text{Softmax}}(\boldsymbol{s}, v^*)$ as the loss of keypoint localization (§3.1). To localize all 14 keypoints, instead of having a separate CNN for each of the keypoint, we use a single CNN that outputs scores $\boldsymbol{s}^k$ for each of the 14 keypoints. This design forces the model to localize all the keypoints jointly, and will learn to infer the location of occluded keypoint based on the locations of other keypoints. The total loss of pose estimation is the sum of the Softmax loss of all 14 keypoints:

$$L_{\text{pose}} = \sum_k L_{\text{Softmax}}(\boldsymbol{s}^k, v^{k^*}), \tag{4}$$

where the index k refers to a particular keypoint. Once the model is trained, it can predict the location of each keypoint k as the voxel with the highest score:

$$\hat{v}_k = \arg\max_v \boldsymbol{s}_v^k. \tag{5}$$

CNN architecture: To localize keypoints in 3D space, our CNN model needs to aggregate information over space to analyze all RF reflections from a person's body and assign scores for each voxel. Also the model needs to aggregate information across time to infer keypoints that may be occluded at a specific time instance. Thus, as illustrated in Fig. 4, our CNN model takes 4D RF tensors (space and time) as input and performs 4D convolution at each layer to aggregate information along space and time, that is:

$$\boldsymbol{a}^n = \boldsymbol{f}^n *_{(4D)} \boldsymbol{a}^{n-1}, \tag{6}$$

where $\boldsymbol{a}^n$ and $\boldsymbol{a}^{n-1}$ are the feature maps at layer n and $n-1$, $\boldsymbol{f}^n$ is the 4D convolution filter at layer n and $*_{4D}$ is 4D convolution operator.

5.2 Challenge: Time and Space Complexity

The 4D CNN model described in §5.1 has practical issues. The time and space complexity of 4D CNN is so prohibitive that major machine learning platforms (PyTorch, Tensorflow) only support convolution operation up to 3D. To appreciate the computational complexity of such model, consider performing 4D convolutions on our 4D RF tensor. The size of the convolution kernel is fixed and relatively small. So the complexity stems from convolving with all 3 spatial dimensions and the time dimension. Say we want to span an area of 100 square meters with 3 meters of elevation. We want to divide this area to voxels of 1 cm^3 to have a good resolution of the location of a keypoint. Also say that we take a time window of 3 seconds and that we have 30 RF measurements per voxel per second. Performing a 4D convolution on such tensor involves $1,000 \times 1,000 \times 300 \times 90$, i.e., 27 giga operations. This process has to be repeated for each example in the training set, which contains over 1.2 million (§8.3) such examples. The training can take multiple weeks. Furthermore, the inference process cannot be performed in real-time.

In fact, the above analysis underestimates the required training and inference time since 4D convolution is one out of multiple high-complexity computations needed by a 4D CNN. Claim 1 below states the complexity of our 4D CNN, which depends on three equally complex computations: 4D convolution (Eqn. 6), Softmax loss computation (Eqn. 4) and maximum score selection (Eqn. 5).

Claim 1. *Assuming the time and space complexity of computing the response of a 4D filter at a single location and time is $O(1)$, the time and space complexity of each 4D convolution, Softmax loss computation and maximum score computation are all $O(XYRT)$, where X, Y, R, T are the size of the input 4D RF tensor along the space and time axes.*

5.3 Model Decomposition

We present a model decomposition that allows us to reduce the complexity from $O(XYRT)$ to $O(XRT + YRT)$. For scenarios in which a resolution of a couple of centimeters is desirable for a space that spans 10×10 square meters, this decomposition translates to 3 orders of magnitude reduction in computation time. In §8.6 we show that such a reduction allows us to infer the 3D skeleton in real-time on a single GPU.

At a high-level, our model decomposition goes as follows: We first prove that our 4D RF tensor is planar decomposable (planar decomposition defined later in Definition 2 and 3). Then we prove that for a layer in a CNN, if its input is planar decomposable, its output is also planar decomposable. Thus, we can stack many convolution layers creating a deep CNN while maintaining decomposability. Finally, we prove that the computation of the loss function and the process of detecting which class has the maximum score are both decomposable when given a decomposable tensor as input. This last step means that we can train the network

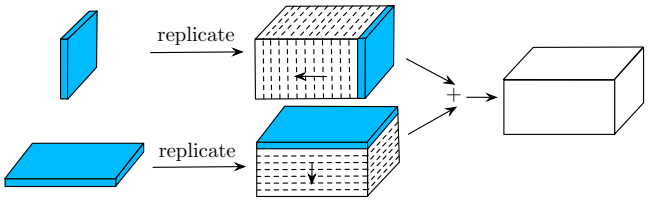


Figure 5: Illustration of Planar Summation.

(and use it for inference) while operating on its decomposed version –i.e., the two 2D planar tensors and the time axis. This completes our model decomposition. Below, we define planar decomposition and state the theorem underlying the model decomposition process, and leave the proofs to the Appendix.

We first define the concept of a planar summation. This concept allows us to create a 3D tensor from two 2D tensors simply by replicating and summing their entries, as shown in Fig. 5. Specifically:

Definition 2 (Planar Summation). *If $\boldsymbol{A}$ is an $n \times l$ matrix and $\boldsymbol{B}$ is an $m \times l$ matrix, then the planar sum $\boldsymbol{A} \oplus \boldsymbol{B}$ is an $n \times m \times l$ 3D tensor $\boldsymbol{C}$, where $\boldsymbol{C}_{i,j,k} = \boldsymbol{A}_{i,k} + \boldsymbol{B}_{j,k}$.*

Analogously, we can define planer decomposition as taking a 3D tensor and decomposing it to two 2D tensors that can regenerate the original 3D tensor using planar summation. Specifically:

Definition 3 (Planar Decomposition). *An $n \times m \times l$ 3D tensor $\boldsymbol{C}$ is planar decomposable if it can be written into the planar sum of an $n \times l$ matrix $\boldsymbol{A}$ and an $m \times l$ matrix $\boldsymbol{B}$, that is, $\boldsymbol{C} = \boldsymbol{A} \oplus \boldsymbol{B}$.*

Once we have defined planar summation and decomposition, the process of decomposing our 4D CNN becomes simple.

(1) First we decompose the RF input.

Theorem 4 (Decomposition of 4D RF tensor by decomposing its spatial dimensions). *The 3D RF tensor from an FMCW array radio with a horizontal array and a vertical array (§3.2) is planar decomposable. It can be decomposed into the planar summation of the 2D RF tensors computed separately from the horizontal array and the vertical array.*

(2) Then, we show that for every convolution layer, if its input is decomposable, its output is also decomposable.

Theorem 5 (Decomposition of Convolution). *For a decomposable 4D tensor $\boldsymbol{A} = \boldsymbol{H} \oplus \boldsymbol{V}$, the output of convolving $\boldsymbol{A}$ with a 4D filter $\boldsymbol{f}$ is also decomposable. That is, there exist 3D tensors $\boldsymbol{H}'$ and $\boldsymbol{V}'$, such that $\boldsymbol{H}' \oplus \boldsymbol{V}' = (\boldsymbol{H} \oplus \boldsymbol{V}) *_{(3D)} \boldsymbol{f}$.*

(3) Next, we show that the loss function and the identification of the class with the maximum score are decomposable. Hence, allowing us to perform training and inference on the decomposed networks.

Theorem 6 (Decomposition of Softmax Loss). *For an $n \times l$ matrix $\boldsymbol{H}$ and an $m \times l$ matrix $\boldsymbol{V}$, Softmax loss $L(\boldsymbol{H} \oplus \boldsymbol{V}, (x^*, y^*, r^*))$ can be computed as:*

$$\log\left(\sum_r \left(\sum_x e^{\boldsymbol{H}_{x,r}}\right) \cdot \left(\sum_y e^{\boldsymbol{V}_{y,r}}\right)\right) - \boldsymbol{H}_{x^*,r^*} - \boldsymbol{V}_{y^*,r^*}$$

Theorem 7 (Decomposition of Maximum Score). *For an $n \times l$ matrix $\boldsymbol{H}$ and an $m \times l$ matrix $\boldsymbol{V}$, the maximum value of $\boldsymbol{H} \oplus \boldsymbol{V}$ can be computed as follows:*

$$\max\left(\boldsymbol{H} \oplus \boldsymbol{V}\right) = \max_r \left(\boldsymbol{h}_r + \boldsymbol{v}_r\right)$$

where $\boldsymbol{h}_r = \max_x \left(\boldsymbol{H}_{x,r}\right)$ and $\boldsymbol{v}_r = \max_y \left(\boldsymbol{V}_{y,r}\right)$.

(4) Finally, Claim 8 below states our final result of reducing the 4D CNN complexity from $O(XYRT)$ to $O(XRT + YRT)$, which is derived directly from the above theorems.

Claim 8. *Assuming the time and space complexity of computing the response of a 4D filter at a single location and time are both $O(1)$, the time and space complexity of each 4D convolution (Theorem 5), Softmax loss computation (Theorem 6) and maximum value computation (Theorem 7) are all $O(XRT + YRT)$, where X, Y, R, T are the size of input 4D RF tensor on spatial and time axis.*

6 MULTI-PERSON 3D POSE ESTIMATION

While the CNN described in the last section can handle single-person 3D pose estimation, the RF signal is capable of capturing multiple people at the same time. Therefore, it is desirable to extend the CNN model so that it can extract 3D skeletons of multiple people from the RF signal. To this end, we follow the divide-and-conquer paradigm by first detecting people regions and then zooming into each region to extract 3D skeleton for each individual. This leads to the design of a new neural network module called region proposal network (RPN), which generates potential people regions.

The most straight-forward approach would have the RPN operate directly on the RF input to identify the 3D region in space that is associated with each person. We can then run our CNN pose-estimation model from last section on the 4D RF tensor after cropping it according to the RPN output. We actually take a different approach: We split the CNN model from last section and make the RPN operate on the output of an intermediate layer (i.e., a feature map), as shown in Fig 6. This approach is inspired by object detection in images; instead of trying to detect objects in the original image, it is preferable to detect objects at an intermediate layer after the information has been condensed. For our application, the

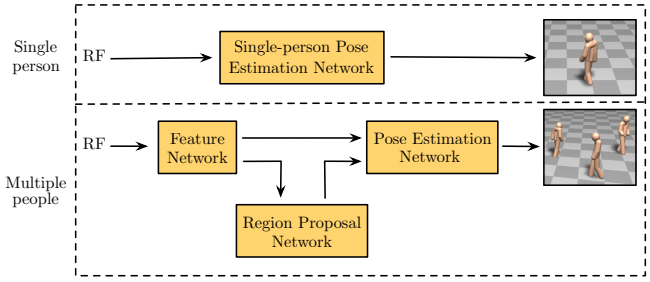


Figure 6: Extension single-person model to multiple people. The single-person pose estimation network is split into feature network and pose estimation network. Critically, region proposal network is inserted to detect individual person based on the output of FN. The skeleton of each individual is further estimated by pose estimation network.

reason why we crop the region associated with a person at an intermediate layer is twofold. First, the raw RF signal is cluttered and suffers from multipath effect. So we want to use a few convolutions layers to condense the information and remove clutter before asking the RPN to crop a specific region (Fig. 12). Second, when multiple people are present, they may occlude each other from the RF device, resulting in missing reflections from the occluded person. Thus we want to perform a few 4D spatio-temporal convolutions to combine information across space and time to allow the RPN to detect a temporarily occluded person

The RPN is inserted in the middle as shown in Figure 6. The CNN model is split into two parts, which we name as feature network (FN) and pose estimation network (PEN). Feature network extracts abstract and high-level feature maps from raw RF signals. Based on these features maps, we first detect potential person regions with RPN. For each region detected by RPN, we zoom into the corresponding region on the feature maps, crop the features and feed them into our pose estimation network.

The single person network contains 18 convolutional layers totally. We split the first 12 layers into feature network (FN) and the remaining 6 layers into pose estimation network (PEN). Where to split is not unique, but generally the FN should have enough layers to aggregate spatial and temporal information for the subsequent RPN and PEN.

6.1 Region Proposal Network

Region proposal network (RPN) is built to generate possible person regions for the subsequent pose estimation network. Ideally a region is a cuboid which tightly bounds a person. While it is inefficient to search over the 3D space to propose such cuboids, it is quite unlikely that one person stands over the head of another person. Therefore, we simplify the 3D cuboid detection as 2D bounding box detection on the horizontal plane (recall that we have decomposed our 4D convolutions to two 3D convolution over horizontal and vertical planes and the time axis).

The RPN takes as input feature maps output by the FN, and outputs a set of rectangular region proposals, each with a score describing the probability of the region containing a person. The RPN is implemented as a standard CNN. One way to train the RPN is to try to all possible regions, and for each region classify it as correct if it fits tightly around a real person in the scene. This approach is very slow since there are so many possible regions. Instead we sample potential regions using a sliding window. For each sampled window, we use a classifier to check whether it intersects reasonably well with a real person. If it does, RPN tries to adjust the boundaries of that window to make it fit better.

We assign a binary label to each window for training, to indicate whether it contains a person or not. To set the label, we use a simple intersection-over-union (IoU) metric, which is defined as:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{7}$$

Therefore, (1) a window that overlaps more than 0.7 IoU with any ground truth region (i.e., a region corresponding to a real person) is set as positive; (2) a window that overlaps less than 0.3 with all ground truth is set as negative; For other windows which satisfy neither of the above criteria, we simply ignore them during the training stage. For other details, we refer the reader to the literature on selecting regions for object detection in images [24].

7 GENERATING 3D POSE LABELS

To learn 3D skeletons from RF signals, RF-Pose3D needs many training examples –i.e., synchronized 4D RF tensors and the corresponding 3D skeletons. In this section, we describe a subsystem that generates such training examples. This subsystem is designed to satisfy the following requirements:

- **Portable and Passive:** It should be portable so that we can collect pose labels from different environments to make sure that our RF-based model can generalize to new scenes. It should also be passive without requiring people to wear any markers, as opposed to motion capture systems (e.g., VICON [28]) that require every person in the scene to put reflective markers around every keypoint.
- **Accurate and Robust:** It should generate accurate 3D skeletons and localize every keypoint on each person with respect to a global reference frame. It also should be robust to various types of occlusions including self-occlusion, inter-person occlusion and occlusion by furniture or walls. Such data is necessary to enable RF-Pose3D to estimate 3D skeletons from different perspectives despite occlusions.

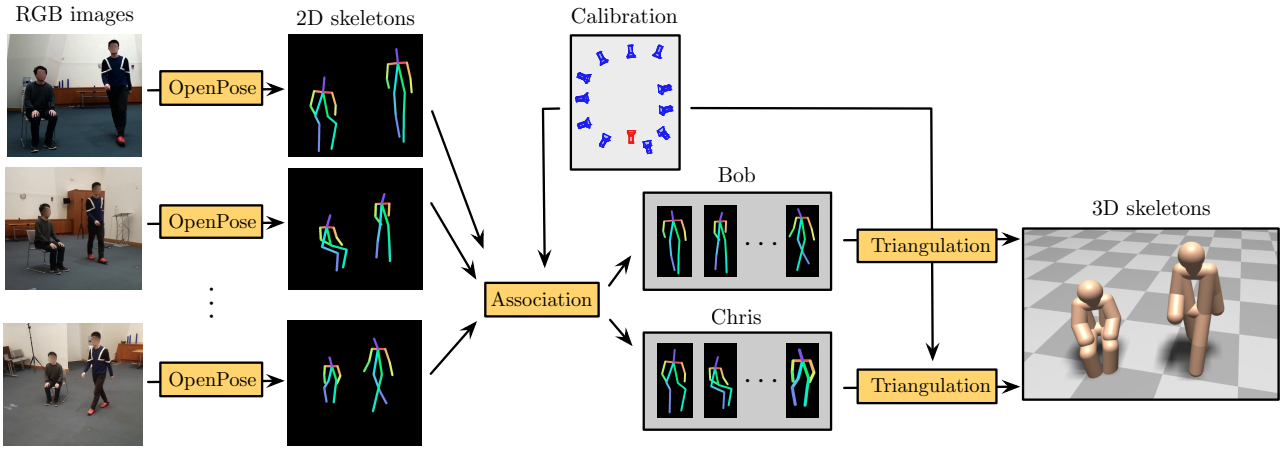


Figure 7: Diagram of 3D skeleton generation using a set of RGB images.

- **Capable of dealing with multiple people:** It should track the 3D skeletons of multiple people simultaneously so that RF-Pose3D has training examples with multiple people and hence can scale to such scenarios.

We have designed and implemented a subsystem for generating labeled examples that satisfy all of the above requirements. Fig. 7 illustrates the operation of this system, which involves the following steps:

Multi-camera system: Our system has 12 camera nodes, each of which consists of a Raspberry Pi, a battery, and a camera module board. Our nodes are small, light, and easy to deploy by attaching them on the wall. The camera nodes are synchronized via NTP and calibrated with respect to one global coordinate system using standard multi-camera calibration techniques [36]. Once deployed, the cameras image people from different view points.

2D skeleton generation: Next, our system uses the images captured by the cameras to generate 2D skeletons. To do so, we leverage a computer vision system called OpenPose [6], which given an image returns the 2D skeletons of the people in it, as shown in Fig. 7. Ideally we would like the same skeletons to appear in the images of all 12 cameras. However, due to occlusions and the fact that 12 cameras are placed to cover different area, each camera may see different people or different keypoints of the same person.

2D skeleton association: Next, we identify 2D skeletons of the same person and associate them together as shown in Fig. 7. To tell whether a pair of 2D skeletons are from the same person or not, we look at the geometric relationship between them. Specifically, given a 2D keypoint (e.g. head), the original 3D keypoint must lie on a line in the 3D space that is perpendicular to the camera view and intersects it at the 2D keypoint. The intuition is that when a pair of 2D skeletons are both from the same person, those two lines corresponding to the potential location of a particular keypoint will intersect in 3D space. On the other hand, if the pair of 2D skeletons are from two different people, those two lines in 3D space will have a large distance and no intersection. Based on this intuition, we use the average distance between the 3D lines corresponding to various keypoints as the distance metric of two 2D skeletons, and use hierarchical clustering [26] to cluster 2D skeletons from the same person.

Triangulating 3D skeletons: Once we have multiple 2D skeletons from the same person, we can triangulate their keypoints to generate the corresponding 3D skeleton. We estimate the 3D location of a particular keypoint $\boldsymbol{p}$ using its 2D projections $\boldsymbol{p}^i$ as the point in space whose projection minimizes the sum of distances from all such 2D projections, i.e.:

$$\boldsymbol{p} = \arg\min_{\boldsymbol{p}} \sum_{i \in I} \left\| C_i \boldsymbol{p} - \boldsymbol{p}^i \right\|_2^2, \tag{8}$$

where the sum is over all cameras that detected that keypoint, and C_i is the calibration matrix that transforms the global coordinates to the image coordinates in the view of camera i [12].

8 IMPLEMENTATION AND EVALUATION

In this section, we describe our implementation, dataset and evaluation results.

8.1 Implementation

Neural Network Architecture. Today there are a few standard CNN designs that are widely used across tasks, we choose to use the ResNet [14] design that uses residual connections across different layers. For more detail about ResNet, please refer to [14]. Our feature network uses a ResNet with 12 layers. Our region proposal network and pose estimation network have another 2 and 6 layers on top of the feature network, respectively. All convolutional layers have a kernel

size of 5 except the region proposal network where the kernel sizes are 3 and 1 for the first and second layer, respectively.

Training Details. All 3 subnetworks are trained jointly using ADAM optimizer [18] with a learning rate of 0.001. Both Residual Connection and Batch Normalization are adopted to benefit the training. To stabilize the training, we balance the loss weights between RPN and PEN as 1 and 0.3, respectively. We use RoiAlign [13] to crop and resize feature maps inside each region proposal.

Camera System. We have implemented a wireless camera system consisting of 12 camera nodes. Each camera node is built on a Raspberry Pi 3 single-board computer resulting in a small box design ($10 \times 7 \times 5cm$) with a light weight ($290g$).

RF Radio. RF-Pose3D uses an FMCW radio equipped with a vertical and horizontal antenna arrays, similar to the one used in [1]. The radio transmits an FMCW chirp sweeping the frequencies from 5.4 to 7.2 GHz. The transmission power is less than one millie Watt. The RF signal is processed using standard FMCW and antenna array equations to generate 30 vertical and horizontal heatmaps per second, which are then synchronized with the camera frames.

Synchronization. Our radio and cameras are synchronized using the network time protocol (NTP). When using a local NTP server, the clock synchronization error is less than 1ms on average. During experiments, we timestamp all the RF heatmaps and video frames and synchronize different streams based on their timestamps. We use an FPS of 30 for all the RF and video streams after synchronization.

8.2 Dataset

We have collected a diverse dataset of synchronized 3D skeletons and RF signals. Our dataset has people performing a variety of typical activities including walking, sitting, hand shaking, using mobile device, chatting, waving hands, etc.

- **Scale**: The dataset contains 16 hours of data. This results in 1,693,440 samples of synchronized 3D skeleton frames and 3D RF tensors.
- **Diversity**: Our data is collected from 22 different locations on a university campus including seminar rooms, open spaces, and offices. The average number of people in each frame is 2.3.
- **Accuracy of 3D skeleton labels**: To evaluate the accuracy of 3D skeletons generated by our camera system (§7), we compare the resulting skeletons against a VICON motion capturing system [28]. Table 1 shows the average distance between 3D skeletons from our camera system and from a VICON system. Our 3D skeletons have an average error of 1.1cm and 1.5cm along two axes on the horizontal plane and 0.7cm along the vertical axis. This result suggests that our 3D skeleton generation subsystem is very accurate and can serve as the ground-truth for training our RF-based model. Note that we could not use the VICON room to generate labeled examples for training since it would limit us to only one environment.

Axis	Avg	Hea	Nec	Sho	Elb	Wri	Hip	Kne	Ank
X	1.1	1.7	0.6	0.8	1.3	1.3	1.1	1.1	1.1
Y	0.7	0.4	0.3	0.4	0.9	1.5	0.6	0.9	0.9
Z	1.5	1.5	1.4	1.2	1.5	1.9	1.6	2.1	1.2

Table 1: Average distance between labels from our camera system and labels from a VICON system. The results show high accuracy and hence justify using our camera system as the ground truth for RF-based 3D skeleton estimation.

8.3 3D Pose Estimation Performance

The 3D pose estimation performance is evaluated by comparing the pose predicted from our model with the ground truth from the camera system. We ensure that the data used for testing and training do not include the same environments.

Training/Testing Split: Our dataset is split into 12 and 4 hours for training and testing, respectively. Our model is trained with data from 16 environments and tested in the remaining 6 environments that are not in the training set.

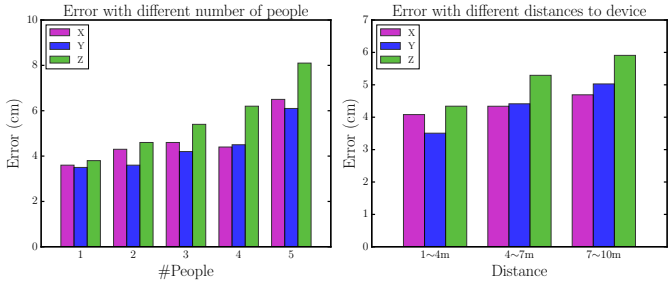
Metric: The spatial distance for each human keypoint between the model predictions and ground truth.

Axis	Avg	Hea	Nec	Sho	Elb	Wri	Hip	Kne	Ank
X	4.2	3.9	3.1	3.6	4.3	5.8	3.2	4.0	5.1
Y	4.0	4.4	4.2	4.3	4.0	5.1	3.5	3.3	3.5
Z	4.9	4.8	3.9	4.4	5.0	6.6	3.8	4.2	5.7

Table 2: Average keypoint localization error (cm) of RF-based 3D skeleton prediction on the test set.

Overall Performance: The keypoint localization performance of our model is shown in Table 2, where the X and Z axes define the horizontal plane and Y is the vertical axis. The average error in localizing a keypoint are 4.2, 4.0 and 4.9 cm in the X, Y and Z axes, respectively. The error along X and Z axes are larger than that of Y axis because of the larger variation of locations in the horizontal plane.

The table reports the localization accuracy for every keypoint type. It merges results for the left and right sides of the body. Evaluated keypoints include head, neck, shoulder, elbow, wrist, hip, knee, and ankle. The results show that our model achieves less error when localizing large or slow body parts, e.g., head or hip, than when localizing small and highly mobile parts, e.g., wrist or ankle. For example, the average error along X, Y and Z when localizing someone's head is 4.4cm, whereas the error in localizing their wrist is 5.8cm. This is expected and can be explained by two reasons. First,



(a) (b)

Figure 8: Keypoint localization error (cm) with (a) different number of people; (b) people at different distances.

the amount of RF reflections highly depends on the size of a body part. Second, limbs such as wrist and ankles are more flexible and their movements usually have a larger degree of freedom than head or hip, thus are harder to be captured.

Overall the accuracy is significantly higher than past localization work, though the task is significantly harder since we are localizing small body parts. This may come as a surprise to some readers. The reason however is threefold. First, a neural network model is much more powerful than a manually crafted model because it can capture dependencies that are unknown to the designer. Second, our model not only captures the information in the RF signal but also the general constraints on the shape and relationship between different body parts. This is because it is trained with many 3D skeletons and hence learns to abstract the relationship between their keypoints. Third, we operate over time and space. Thus, the model can learn the dynamics of how each keypoint moves and use the information to predict the location of a keypoint even when it is occluded.

Figure 9: Through wall example. The top left image represents the view of the radio, the top right image shows the view inside the room. Bottom row shows the detected skeletons in corresponding views.

Different Number of People: The performance on different number of people is reported in Fig. 8(a). The average error along the spatial dimensions for a single person is 3.8cm. As the number of subjects goes to 5, the average error increases to 7cm, which is caused by heavy inter-people occlusion. The ability to sustain such accuracy with multiple people is due to our RPN module, which can zoom in on each person and reduce interference from other people and the environment. One major reason we do not train with more than 5 people is that the camera system starts to become unstable due to heavy occlusions. Potentially our model can be trained and tested with more people if a better camera system is constructed to provide supervision (for example by increasing the number of coordinated cameras).

Different Ranges: We evaluate the performance when people are located at different distances. Fig. 8(b) shows that as people move from 1m to 10m, the error slightly increases from 3.8cm to 5.3cm. The increase in error is expected since the spatial resolution of antenna arrays decreases with distance (an angular error of a few degrees leads to small errors at nearby distances but large errors at far distances.) We did not experiment with distances larger than 10 meters because at such distances the main limitation is the low power of the FMCW radio [2].

Same v.s. Different Environment: All of the above results were for training and testing on different environment. In this section, we train and test our model in the same environment in order to compare with cross environment testing result. Note that though we use the same environment, we still use different examples for training and testing. The average error along X, Y, and Z is 3.7cm which is on par with cross environment error which is 4.4cm. This clearly shows that our model is robust to environmental changes. Again, this benefit stems from the RPN module which enables the PEN to focus on individual people and ignore environmental reflectors.

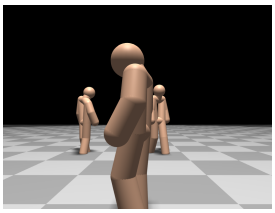
Through-Wall v.s. Line of Sight: We evaluate our system in through-wall scenarios where the radio is separated from the monitored people by a wall. The errors along the X, Y and Z axes are 5.2cm, 3.7cm and 4.7cm, respectively. These errors are comparable to the errors in line-of-sight scenarios which are reported in Table 2. One example is shown in Fig. 9, where the top left image shows the viewpoint of the radio, the right image shows the view inside the room. The second row shows the 3D skeletons from the corresponding viewpoints.

Qualitative Results: Fig. 10 and Fig. 11 shows samples of 3D skeletons for multiple people generated using RF-Pose3D. It illustrates that RF-Pose3D works well in different environments and when people are doing a variety of activities, e.g., sitting, walking, interacting with each other, etc.

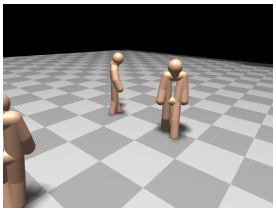
Figure 10: Qualitative results on multi-person detection and pose estimation.

(a) 1st Camera View

(b) 2nd Camera View



(c) 1st Prediction View



(d) 2nd Prediction View

Figure 11: RF-Pose3D generates 3D skeletons from different perspectives. Top row shows two views out of the camera system, bottom row shows the detected skeletons in corresponding views.

8.4 Performance of Human Detection

Recall that our model starts by detecting people and zooming on each of them to extract his or her skeleton. Thus we evaluate the human detection performance of our model–i.e., whether it correctly detects all the people in the environment despite fictitious people due to multipath or other objects.

Metrics: We use the following metrics that are commonly used in object detection tasks.

- *Precision:* Precision is defined as the fraction of detected regions that truly contain a person. It measures the robustness of our system against false positives, i.e., fictitious people.
- *Recall:* Recall is defined as the fraction of people that are detected over the total amount of people. It measures our system's ability in detecting all the people without misses.
- *F1 score:* F1 considers both precision and recall, and is computed as the harmonic average of the two, i.e., $\frac{2 \cdot p \cdot r}{p+r}$.

Table 3 shows the precision and recall for test data with different number of people in the scene. Overall, our model achieves a precision of 95.8% and a recall of 99.6% on single-person data. As the number of people increases, the F1 score only drops slightly by 2.9%. This demonstrates the effectiveness of our region proposal network, which successfully detects multiple people in the environment without being fooled by multipath or objects in the environment. This is partly attributed to the feature network which learns to attenuate the side effect of multipath as well as aggregate beneficial temporal information.

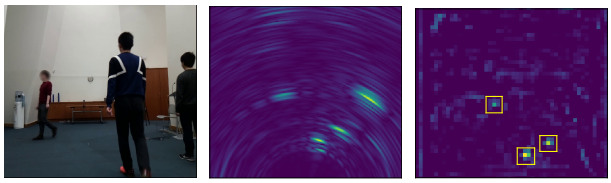
#People	1	2	3	4	5
Precision (%)	95.8	96.2	94.9	95.2	96.3
Recall (%)	99.6	98.8	97.8	96.5	93.4
F-1 score	97.7	97.4	96.3	95.9	94.8

Table 3: Precision and Recall when there are different number of people.

To better understand how RPN works consider the example in Fig. 12. The left part of Fig. 12 shows an experiment where there are three people in the scene. The middle part of the figure shows the horizontal RF tensor at that instance of time, which contains multipath reflections from the wall. The right part of the figure shows one of the feature maps from the feature network together with the regions proposed by RPN. The feature map has a large value only at the locations of the three people, suggesting that the feature network has learned to differentiate reflections from real people from fictitious ones due to multipath and objects in the environment. In this example, the RPN successfully detects all the people and have 3 proposal boxes corresponding to each of them.

8.5 Localization Performance

We also compare our model with past work on indoor localization. Our trained model can derive people's location



(a) RGB image **(b)** Horizontal RF tensor **(c)** Feature map

Figure 12: An example output of RPN. Left: RGB images from the view of the device. Middle: horizontal RF tensor that contains fictitious people along with real ones. Right: Decomposed horizontal feature map after FN, marked with detected regions. Fictitious people are removed, real ones are detected.

Methods	Median			90-th Percentile		
	X	Y	Z	X	Y	Z
RF-Pose3D	1.7	2.8	2.3	5.1	8.3	6.4
WiTrack [2]	9.9	8.6	17.7	35.0	20.0	60.0

Table 4: Comparison with previous device-free localization system. Median and 90-th percentile localization error (cm) of RF-Pose3D and WiTrack.

Model	FN	RPN	PEN	Total	4D CNN
Time (s)	0.04	0.01	0.34	0.39	87.0 (estimated)

Table 5: Runtime analysis of our model during inference on a single NVIDIA Titan X GPU. The table shows the time spent on each part of our model for every 1 second of RF signal. It suggests that our model can perform inference in real-time with our decomposition techniques while a vanilla 4D CNN could take 87.0 seconds from estimation.

simply by computing the center of neck, two shoulders and two hips. We compare our method with previous RF-based device-free indoor localization system WiTrack [2] in Table 4. Our system achieves a median error of 1.7, 2.8 and 2.3 on X, Y and Z axes, respectively and 90-th percentile error of 5.1, 8.3 and 6.4, which is significantly better than past localization systems. This results demonstrates the power of the new model and the importance of the extra information it can get from the wireless signal even for more traditional tasks like localization.

8.6 Running Time Analysis

As explained in §5.3, the proposed planar tensor decomposition technique enables us to train and test on 4D tensor data using 3D convolutions. Here, we provide a quantitative analysis of it. In Table 5, we benchmark the inference runtime of the three subnetworks of our model: FN, FPN and PEN. On a single NVIDIA Titan X GPU, one second of RF tensor data takes only 0.39 seconds to process. Estimated from the number of floating point operations, it would take a 4D CNN approximately 87 seconds to perform inference, which is way below real-time.

9 DISCUSSION

We present RF-Pose3D, a device-free system that for the first time estimates 3D human skeletons from RF signals. By designing a novel CNN model and leveraging camera system for supervision, RF-Pose3D is able to detect 3D skeletons for multiple people simultaneously. In terms of modeling, to avoid high dimensional convolution operations, we propose a tensor decomposition technique that is computationally efficient, making the system capable of running in realtime.

RF-Pose3D provides a leap in the quality and richness of human-related information learned from RF signals. However, the system exhibits some limitations: First, our dataset is focused on common activities in office buildings (e.g., walking, sitting, standing) and misses certain poses, e.g., dancing and doing sports. As a result, the trained model is good for poses common in office buildings and may degenerate with poses it did not see in the dataset. This problem can be addressed by expanding the dataset to include more actions. Second, the radio we use in this paper can work up to 40 feet. Extra transmission power or multiple radios would be needed in order to cover a larger space. Third, the efficacy of RF-based pose estimation depends on the power reflected from each body part. Naturally, smaller body parts (e.g., hands and wrists) reflect less power than larger ones. Thus, learning actions that involve complex hand motion is more difficult. Despite these limitations, we see this paper as an important step towards using wireless signals for human sensing. We believe this non-contact 3D pose tracking system can enable new applications in healthcare, smart homes and video gaming.

ACKNOWLEDGMENTS

We thank our shepherd, Lili Qiu, and the anonymous reviewers for their comments and feedback. We also thank all the human subjects for their contribution to our dataset. The authors are grateful to the NETMIT members for discussion and support.

A PROOF OF THEOREMS

Proof of Theorem 4. We prove that each 3D RF tensor is planar decomposible, and therefore the 4D RF tensor (3D RF tensor over time) is also planar decomposible. Consider an FMCW array with M and N receivers for the horizontal and vertical arrays, respectively. Let (x, y, r) denotes 3D location in the (X, Y, R)-coordinate system as shown in Fig. 13, where r is the distance from the point (x, y, r) to the origin. Let $d_m^h(x, y, r)$ denotes the round trip distance from transmit antenna to the point at the 3D voxel at (x, y, r) and back to the m-th horizontal receive antenna. $d_n^v(x, y, r)$ is similiarly defined for the n-th vertical receive antenna.

Base on Eqn. 3, the 3D RF tensor is computed as:

$$A(x,y,r) = \sum_{m=1}^{M}\sum_{i} s^{h}_{m,i}\cdot e^{j2\pi\frac{d^{h}_{m}(x,y,r)}{\lambda_i}} + \sum_{n=1}^{N}\sum_{i} s^{v}_{n,i}\cdot e^{j2\pi\frac{d^{v}_{n}(x,y,r)}{\lambda_i}}$$

and the 2D RF tensor based on horizontal and vertical array are computed as:

$$H(x,r) = \sum_{m=1}^{M}\sum_{i} s^{h}_{m,i}\cdot e^{j2\pi\frac{d^{h}_{m}(x,0,r)}{\lambda_i}}$$

$$V(y,r) = \sum_{n=1}^{N}\sum_{i} s^{v}_{n,i}\cdot e^{j2\pi\frac{d^{v}_{n}(0,y,r)}{\lambda_i}}$$

It can be proved geometrically that $d^h_m(x,y,r) = d^h_m(x,0,r)$ and $d^v_n(x,y,r) = d^v_n(0,y,r)$, therefore $A(x,y,r) = H(x,r) + V(y,r)$, that is $A = H \oplus V$. □

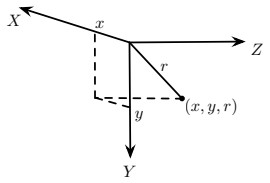


Figure 13: (X,Y,R)-coordinate system.

Proof of Theorem 5. Due to space limit, we only prove the decomposition of 3D spatial convolution below, and 4D convolution is a natural extension of it. For an $n \times l$ matrix $\boldsymbol{H}$ and an $m \times l$ matrix $\boldsymbol{V}$, we prove that: $(\boldsymbol{H} \oplus \boldsymbol{V}) *_{(3D)} (\boldsymbol{f}^h \oplus \boldsymbol{f}^v) = \boldsymbol{H'} \oplus \boldsymbol{V'}$, where $\boldsymbol{H'} = (m\boldsymbol{H} + \boldsymbol{1}_{n\times m}\cdot \boldsymbol{V}) *_{(2D)} \boldsymbol{f}^h$, $\boldsymbol{V'} = (n\boldsymbol{V} + \boldsymbol{1}_{m\times n}\cdot \boldsymbol{H}) *_{(2D)} \boldsymbol{f}^v$, and $\boldsymbol{1}_{a\times b}$ is a-by-b all-one matrix. Let $A = (\boldsymbol{H} \oplus \boldsymbol{V}) *_{(3D)} (\boldsymbol{f}^h \oplus \boldsymbol{f}^v)$.

$$\boldsymbol{A}(x,y,r) = \sum_{i,j,k} \big(\boldsymbol{H}(x+i, r+k)+\boldsymbol{V}(y+j, r+k)\big)\cdot\big(\boldsymbol{f}^h(i,k)+\boldsymbol{f}^v(j,k)\big)$$

$$\boldsymbol{H'}(x,r) = m\sum_{i,k}\boldsymbol{H}(x+i, r+k)\boldsymbol{f}^h(i,k) + \sum_{i,k}\sum_{j}\boldsymbol{V}(j, r+k)\boldsymbol{f}^h(i,k)$$

$$\boldsymbol{V'}(y,r) = n\sum_{j,k}\boldsymbol{V}(y+j, r+k)\boldsymbol{f}^v(j,k) + \sum_{j,k}\sum_{i}\boldsymbol{H}(i, r+k)\boldsymbol{f}^v(j,k)$$

It can be examined that $\boldsymbol{A}(x,y,r) = \boldsymbol{H'}(x,r) + \boldsymbol{V'}(x,r)$, hence $\boldsymbol{A} = \boldsymbol{H'} \oplus \boldsymbol{V'}$ by definition. □

Proof of Theorem 6. Base on Eqn. 1:

$$L(\boldsymbol{H}\oplus\boldsymbol{V}, (x^*,y^*,r^*)) = \log\Big(\sum_{x,y,r} e^{\boldsymbol{H}_{x,r}+\boldsymbol{V}_{y,r}}\Big) - \boldsymbol{H}_{x^*,r^*} - \boldsymbol{V}_{y^*,r^*}$$

$$= \log\Big(\sum_{r}\Big(\sum_{x} e^{\boldsymbol{H}_{x,r}}\Big)\cdot\Big(\sum_{y} e^{\boldsymbol{V}_{y,r}}\Big)\Big) - \boldsymbol{H}_{x^*,r^*} - \boldsymbol{V}_{y^*,r^*}$$

Proof of Theorem 7.

$$\begin{aligned}\max(\boldsymbol{H}\oplus\boldsymbol{V}) &= \max_{x,y,r}(\boldsymbol{H}_{x,r}+\boldsymbol{V}_{y,r})\\ &= \max_{r}\big(\max_{x}\boldsymbol{H}_{x,r} + \max_{y}\boldsymbol{V}_{y,r}\big)\\ &= \max_{r}(\boldsymbol{h}_r + \boldsymbol{v}_r)\end{aligned}$$

REFERENCES

[1] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. *ACM Transactions on Graphics* 34, 6 (November 2015), 219.

[2] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D tracking via body radio reflections. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)*, Vol. 14. 317–329.

[3] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C Miller. 2015. Smart homes that monitor breathing and heart rate. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*.

[4] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. 2016. Soundnet: Learning sound representations from unlabeled video. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*. 892–900.

[5] Petr Beckmann and Andre Spizzichino. 1987. *The scattering of electromagnetic waves from rough surfaces.* Pergamon Press.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7291–7299.

[7] Human3.6M Dataset. 2018. http://vision.imar.ro/human3.6m (accessed January 31, 2018).

[8] Haoshu Fang, Shuqin Xie, and Cewu Lu. 2017. RMPE: Regional multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2334–2343.

[9] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1440–1448.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.

[11] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. 2014. Using k-poselets for detecting people and localizing their keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3582–3589.

[12] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision.* Cambridge University Press.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2980–2988.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Chen-Yu Hsu, Yuchen Liu, Zachary Kabelac, Rumen Hristov, Dina Katabi, and Christine Liu. 2017. Extracting gait velocity and stride length from surrounding radio signals. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI)*. 2116–2126.

[16] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. 2016. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 34–50.

[17] Kiran Raj Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. 2015. WiDeo: Fine-grained device-free motion tracing using RF backscatter. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)*. 189–204.

[18] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[19] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using WiFi. In *ACM SIGCOMM Computer Communication Review*. 269–282.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*. 1097–1105.

[21] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. 2014. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 541–551.

[22] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. 2016. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4929–4937.

[23] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th Annual International Conference on Mobile computing & Networking (MobiCom)*. ACM, 27–38.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*. 91–99.

[25] Mark A Richards. 2005. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education.

[26] Lior Rokach and Oded Maimon. 2005. Clustering methods. In *Data mining and knowledge discovery handbook*. Springer.

[27] Leonid Sigal, Alexandru O Balan, and Michael J Black. 2010. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision* 87, 1 (March 2010), 4–27.

[28] Vicon Motion Systems. 2018. https://www.vicon.com/ (accessed January 31, 2018).

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 4489–4497.

[30] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*. 613–621.

[31] Wei Wang, Alex X Liu, and Muhammad Shahzad. 2016. Gait recognition using WiFi signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM.

[32] Yuxi Wang, Kaishun Wu, and Lionel M Ni. 2017. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing* 16, 2 (2017), 581–594.

[33] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4724–4732.

[34] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI)*. 71–84.

[35] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. 2016. WiWho: WiFi-based person identification in smart spaces. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks (IPSN)*. 4.

[36] Zhengyou Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (2000), 1330–1334.

[37] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (February 2012), 4–10.

[38] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. 2018. The sound of pixels. *arXiv preprint arXiv:1804.03160* (2018).

[39] Mingmin Zhao, Fadel Adib, and Dina Katabi. 2016. Emotion recognition using wireless signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*.

[40] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[41] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. 2017. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International Conference on Machine Learning (ICML)*.