IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition

HYEOKHYEN KWON[*], School of Interactive Computing, Georgia Tech
CATHERINE TONG[*], Department of Computer Science, University of Oxford
HARISH HARESAMUDRAM, School of Electrical and Computer Engineering, Georgia Tech
YAN GAO, Department of Computer Science, University of Oxford
GREGORY D. ABOWD, School of Interactive Computing, Georgia Tech
NICHOLAS D. LANE, Department of Computer Science, University of Oxford
THOMAS PLÖTZ, School of Interactive Computing, Georgia Tech

The lack of large-scale, labeled data sets impedes progress in developing robust and generalized predictive models for on-body sensor-based human activity recognition (HAR). Labeled data in human activity recognition is scarce and hard to come by, as sensor data collection is expensive, and the annotation is time-consuming and error-prone. To address this problem, we introduce IMUTube, an automated processing pipeline that integrates existing computer vision and signal processing techniques to convert videos of human activity into virtual streams of IMU data. These virtual IMU streams represent accelerometry at a wide variety of locations on the human body. We show how the virtually-generated IMU data improves the performance of a variety of models on known HAR datasets. Our initial results are very promising, but the greater promise of this work lies in a collective approach by the computer vision, signal processing, and activity recognition communities to extend this work in ways that we outline. This should lead to on-body, sensor-based HAR becoming yet another success story in large-dataset breakthroughs in recognition.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; • **Computing methodologies** → **Artificial intelligence**; *Supervised learning by classification.*

Additional Key Words and Phrases: Activity Recognition, Data Collection, Machine Learning

ACM Reference Format:
Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. 2018. IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. 1, 1 (June 2018), 26 pages. https://doi.org/10.1145/1122445.1122456

[*] Both authors contributed equally to this research.

Authors' addresses: Hyeokhyen Kwon, hyeokhyen@gatech.edu, School of Interactive Computing, Georgia Tech, 30332; Catherine Tong, eu.tong@cs.ox.ac.uk, Department of Computer Science, University of Oxford; Harish Haresamudram, harishkashyap@gatech.edu, School of Electrical and Computer Engineering, Georgia Tech; Yan Gao, yan.gao@keble.ox.ac.uk, Department of Computer Science, University of Oxford; Gregory D. Abowd, abowd@gatech.edu, School of Interactive Computing, Georgia Tech; Nicholas D. Lane, nicholas.lane@cs.ox.ac.uk, Department of Computer Science, University of Oxford; Thomas Plötz, thomas.ploetz@gatech.edu, School of Interactive Computing, Georgia Tech.

XXXX-XXXX/2018/6-ART $15.00
https://doi.org/10.1145/1122445.1122456

1 Introduction

On-body sensor-based human activity recognition (HAR) is widely utilized for behavioral analysis, such as user authentication, healthcare, and tracking everyday activities [5, 13, 44, 69, 85]. Regardless of its utility, the HAR field has yet to experience significant improvements in recognition accuracy, in contrast to the breakthroughs in other fields, such as speech recognition [30], natural language processing [17], and computer vision [29]. In those domains it is possible to collect huge amounts of labeled data, the key for deriving robust recognition models that strongly generalize across application boundaries. In contrast, collecting large-scale, labeled data sets has so far been limited in sensor-based human activity recognition. Labeled data in human activity recognition is scarce and hard to come by, as sensor data collection is expensive, and the annotation is time-consuming and sometimes even impossible for privacy or other practical reasons. A model derived from such a sparse dataset is not likely to generalize well. Despite the numerous efforts in improving human activity dataset collection, the scale of typical datasets remains small, thereby only covering limited sets of activities [13, 31, 78, 85]. Even the largest sensor-based activity dataset only spans a few dozen users and relatively short durations [5, 64], which is in stark contrast to the massive datasets in other domains that are often several orders of magnitude larger. For example, Daphnet freezing of gait dataset [5] has 5 hours of sensor data from 10 subjects, and PAMAP2 dataset [64] has 7.5 hours of sensor data from 9 subjects. However, for reference, the "imagenet" dataset [16] has approx. 14 million images, and the "One billion words" benchmark [14] contains one billion words.

In this work, we propose a novel approach that can potentially alleviate the sparse data problem in sensor-based human activity recognition. We aim at harvesting existing video data from large-scale repositories, such as YouTube, and automatically generate data for virtual, body-worn movement sensors (IMUs) that will then be used for deriving sensor-based human activity recognition systems that can be used in real-world settings. The overarching idea is appealing due to the sheer size of common video repositories and the availability of labels in form of video titles and descriptions. Having access to such data repositories opens up possibilities for more robust and potentially more complex activity recognition models that can be employed in entirely new application scenarios, which so far could not have been targeted due to limited robustness of the learned models. The challenges for making these vast amounts of existing data usable for sensor-based activity recognition are manyfold, though: *i)* the datasets needs to be curated and filtered towards the actual activities of interest; *ii)* even though video data capture the same information about activities in principle, sophisticated preprocessing is required to match the source and target sensing domains; *iii)* the opportunistic use of activity videos requires adaptations to account for contextual factors such as multiple scene changes, rapid camera orientation changes (landscape/portrait), scale of the performer in the far sight, or multiple background people not involved in the activity; and *iv)* new forms of features and activity recognition models will need to be designed to overcome the short-comings of learning from video-sourced motion information for eventual IMU-based inference.

We provide a proof-of-concept on how video data can effectively be used for training sensor-based activity recognizers, and as such demonstrate the first step towards larger-scale, and more complex deployment scenarios than what is considered the state-of-the-art in the field. Our approach extracts motion information from arbitrary human activity videos, and is thereby not limited to specific scenes or viewpoints. We have developed **IMUTube**, an automated processing pipeline that: *i)* applies standard pose tracking and 3D scene understanding techniques to estimate full 3D human motion from a video segment that captures a target activity; *ii)* translates the visual tracking information into virtual motion sensors (IMU) that are placed on dedicated body positions; *iii)* adapts the virtual IMU data towards the target domain through distribution matching; and *iv)* derives activity recognizers from the generated virtual sensor data, potentially enriched with small amounts of real sensor data. Our pipeline integrates a number of off-the-shelf computer vision and graphics techniques, so that IMUTube is fully automated and thus directly applicable to a rich variety of existing videos. One notable limitation from our current prototype is that it still requires human curation of videos to select appropriate activity content. However, with advances in computer vision the potential of our approach can be further increased towards complete automation.

The work presented in this paper is the first step towards the greater vision of automatically deriving robust activity recognition systems for body-worn sensing systems. The key idea is to opportunistically utilize as much existing data and information as possible thereby not being limited to the particular target sensing modalities. We present the overall approach and relevant technical details and explore the potential of the approach on practical recognition scenarios. Through a series of experiments on three benchmark datasets—RealWorld [76], PAMAP2 [64], and Opportunity [13]—we demonstrate the effectiveness of our approach. We discuss the overall potential of models trained purely on virtual sensor data, which in certain cases can even reach recognition accuracies that are comparable to models that are trained only on actual sensor data. Moreover, we show that when adding only small portions of real sensor data during model training we are even able to outperform those models that were trained on real sensor data alone. As such, our experiments show the potential of the proposed approach, a paradigm shift for deriving sensor-based human activity recognition systems.

This work opens up the opportunity for the human activity recognition community to expand the general focus towards more complex and more challenging recognition scenarios. We expect the proposed approach to dramatically accelerate the progress of human activity recognition research. With the proposed method it will also be possible to freely experiment with and optimize on-body sensor configurations, which will have substantial impact on real-world deployments. We discuss possible extensions to the presented approach, and thus define a research agenda towards next generation sensor-based human activity recognition.

2 Extracting Virtual IMU Data from Videos

The key idea of our work is to replace the conventional data collection procedure that is typically employed for the development of sensor-based human activity recognition (HAR) systems. Our approach aims at making existing, large-scale video repositories accessible for the HAR domain, leading to training datasets of sensor data, such as IMUs, that are potentially multiple orders of magnitude larger than what is standard today. With such a massively increased volume of *real* movement data—in contrast to simulated or generated samples, that often do not exhibit the required quality nor variability—it will become possible to develop substantially more complex and more robust activity recognition systems with potentially much broader scope than the state-of-the-art in the field. In what follows, we first give an overview of the general approach before we provide the technical details of our procedure that converts videos into virtual IMU data.

2.1 IMUTube Overview

Figure 1 gives on overview of the proposed paradigm shift for deriving sensor-based human activity recognition systems. The top left part ("conventional") summarizes the currently predominant protocol. Study participants are recruited and invited for data collection into a laboratory environment. There they wear the sensing platforms, such as a wrist-worn IMU, and engage into the activities of interest, typically in front of a camera. Human annotators provide ground truth labeling either directly, i.e., while the activities are performed, or based on the video footage from the recording session. This procedure is very labor intensive and often error-prone, and, as such, labeled datasets of only limited size can typically be recorded with reasonable efforts.

In contrast, our approach aims at utilizing existing, large-scale repositories of videos that capture activities of interest (bottom left part of Figure 1 labelled "IMUTube"). With the explosive growth of social media platforms, a virtually unlimited supply of labeled video is available online that we aim to utilize for training sensor-based HAR systems. In our envisioned application, a query for a specific activity delivers a (large) set of videos that seemingly capture the target activity. These results (currently) need to be curated in order to eliminate obvious outliers etc. such that the videos are actually relevant to the task (see discussion in Section 6). Our processing pipeline then converts the video data into usable virtual sensor (IMU) data. The procedure is based on a computer

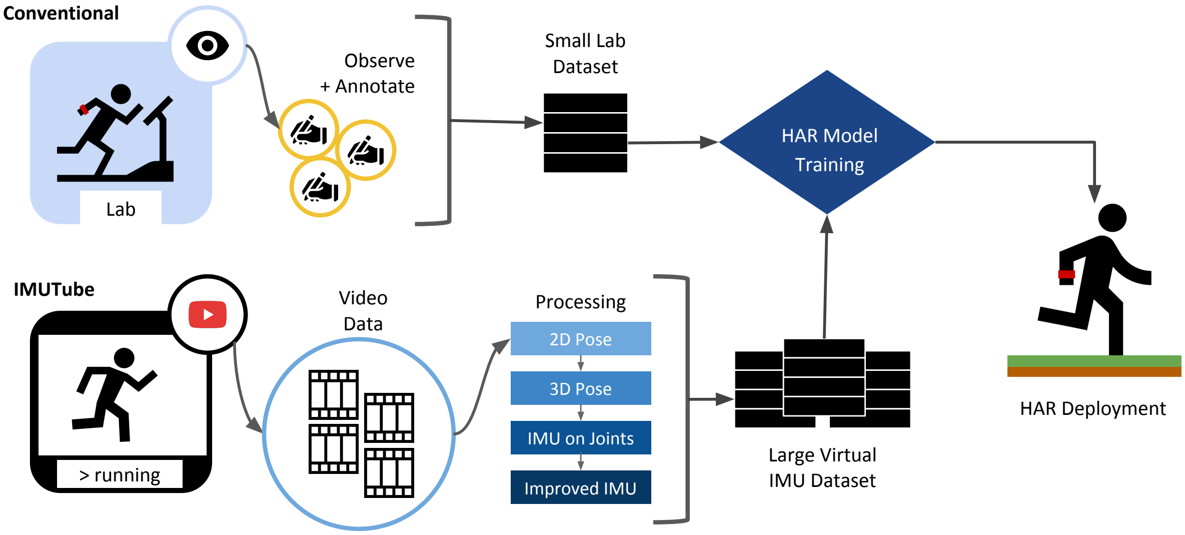


Fig. 1. The proposed IMUTube system replaces the conventional data recording and annotation protocol (upper left) for developing sensor-based human activity recognition (HAR) systems (upper right). We utilize existing, large-scale video repositories from which we generate virtual IMU data that are then used for training the HAR system (bottom part).

vision pipeline that first extracts 2D pose information, which is then lifted to 3D. Through tracking individual joints of the extracted 3D poses, we are then able to generate sensor data, such as tri-axial acceleration values, at many locations on the figure. These values are then post-processed to match the target application domain.

The proposed work aims at replacing the data collection phase of HAR development. It is universal as it does not impose constraints on model training (top center in Figure 1) nor deployment (right part of Figure 1). In what follows we describe the technical details of our processing pipeline that make videos usable for training IMU-based activity recognition systems. This description assumes direct access to a video that captures a target activity, i.e., here we do not focus on the logistics and practicalities of querying video repositories and curating the search results.

2.2 Motion Estimation for 3D Joints

On-body movement sensors capture local 3D joint motion, and, as such, our processing pipeline aims at reproducing this information but from 2D video. As shown in Figure 2 we employ a two step approach. First, we estimate 2D pose skeletons for potentially multiple people in a scene using a state-of-the-art pose extractor, namely the *Openpose* model [10]. Then, each 2D pose is lifted to 3D by estimating the depth information that is missing in 2D videos. Without limiting the general applicability we assume here that all people in a scene are performing the same activity. Although fast and accurate, the *Openpose* model estimates 2D poses of people on a frame by frame basis only, i.e., no tracking is included which requires post-processing to establish and maintain person correspondences across frames. In response, we apply the *SORT* tracking algorithm [7] to track each person across the video sequence. *SORT* utilizes bipartite graph matching with the edge weights as the intersection-over-union (IOU) distance between boundary boxes of people from consecutive frames. The boundary boxes are derived as tight boxes including the 2D keypoints for each person.

To increase the reliability of the 2D pose detection and tracking, we remove 2D poses where over half of the joints are missing, and also drop sequences that are shorter than one second. For each sequence of a tracked

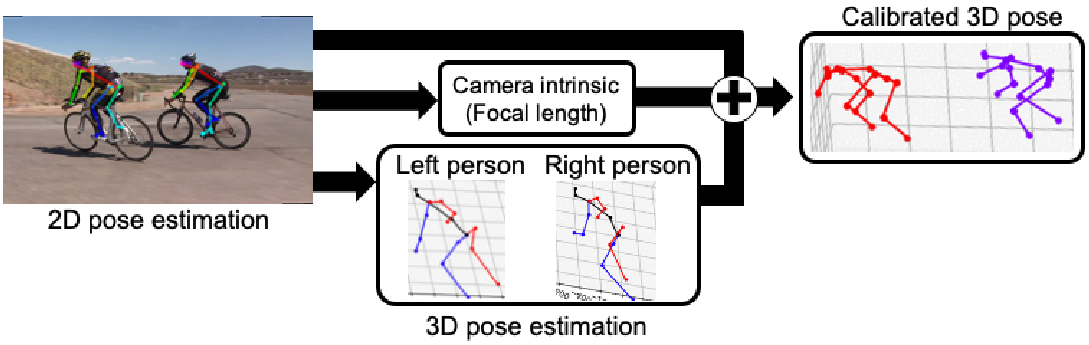


Fig. 2. 3D joint orientation estimation and pose calibration. The multi-person 2D poses are estimated with *Openpose* followed by lifting to 3D through *VideoPose3D*. The camera intrinsic parameters are estimated using the *DeepCalib* model. Jointly with the pose and camera related parameters we calibrate the orientation and translation in the 3D scene for each frame.

person, we also interpolate and smooth missing or noisy keypoints in each frame using a Kalman filter, as pose change cannot be dramatically different within frames. Finally, each 2D pose sequence is lifted to 3D pose by employing the *VideoPose3D* model [56]. Capturing the inherent smooth transition of 2D poses across the frame encourages more natural 3D motion in the final estimated (lifted) 3D pose.

2.3 Global Body Tracking in 3D

Inertial measurement units capture the acceleration from global body movement in 3D, and additionally local joint motions in 3D. Thus, we also have to extract global 3D scene information from the 2D video to track a person's movement in the whole scene. Typical 3D pose estimation models do not localize the global 3D position and orientation of the pose in the scene. Tracking the three-dimensional position of person in 2D videos requires two pieces of information: *i)* 3D localization in each 2D frame; and *ii)* the camera viewpoint changes (ego-motion) between subsequent 3D scenes. We map the 3D pose of a frame to the corresponding position within the whole 3D scene in the video, compensating for the camera viewpoint of the frame. The sequence of the location and orientation of 3D pose is the global body movement in the whole 3D space. For the virtual sensor, the global acceleration from the tracked sequence will be extracted along with local joint acceleration.

2.3.1 3D Pose Calibration First, we estimate the 3D rotation and translation of the 3D pose within a frame, as shown in Figure 2. For each frame, we calibrate each 3D pose from a previously estimated 3D joint according to the perspective projection between corresponding 3D and 2D keypoints. The perspective projection can be estimated with the Perspective-n-point (Pnp) algorithm [33]. Additionally to 3D and 2D correspondences, the Pnp algorithm requires the camera intrinsic parameters for the projection, which include focal length, image center, and the lens distortion parameters [11, 70]. Since arbitrary online video typically do not come with such EXIF metadata, the camera intrinsic parameters are estimated from the video with the *DeepCalib* model [8]. The *DeepCalib* model is a frame-based model that considers a single image at a time so that the estimated intrinsic parameter for each frame slightly differs across the frame according to its scene structure. Hence, we assume that a given video clip sequence is recorded with a single camera, and aggregate the intrinsic parameter predictions by taking the average from all the frames:

$$c^{int} = \frac{1}{T}\sum_{t=1}^{T} c_t^{int} \tag{1}$$

where, $c^{int} = [f, p, d]$ is the averaged camera intrinsic parameters from each frame, x_t at time t, predictions, $c_t^{int} = DeepCalib(x_t)$. $f = [f_x, f_y]$ is the focal length and $p = [p_x, p_y]$ is optical center for x and y axis, and d

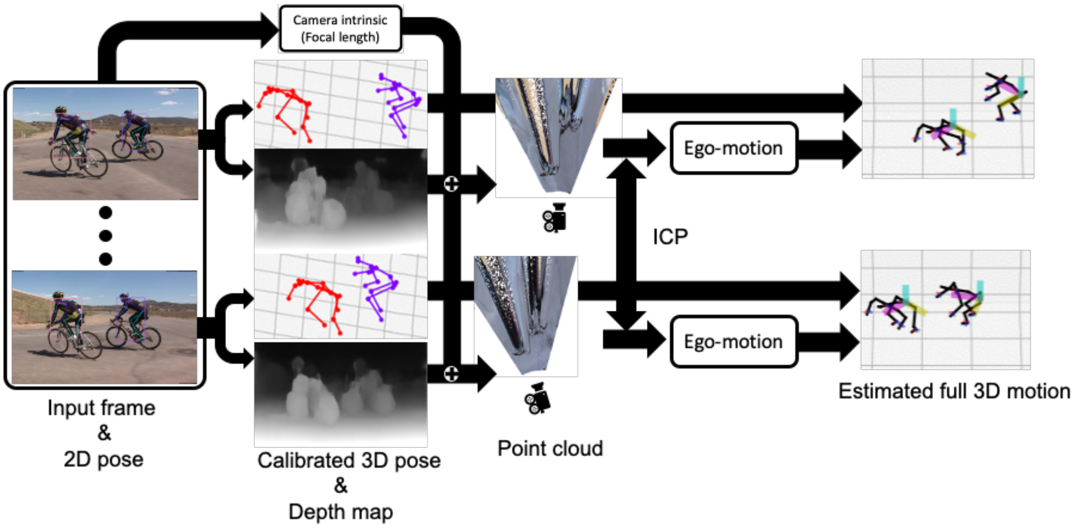


Fig. 3. 3D pose and motion tracking with compensating the camera motion. The camera motion is estimated through the iterative closest point (ICP) algorithm between subsequent pointclouds. Then, calibrated 3D poses per frame are mapped to the location in the entire 3D scene, compensating the camera motion. The calibrated 3D poses from both frames are initially centered in the 3D world origin as the camera follows the cyclists. After incorporating ego-motion information, we can see that two cyclists are moving from right to left, moving closer to each other as in the video (most right figure).

denotes the lense distortion. Once the camera intrinsic parameter is calculated, the Pnp algorithm regresses global pose rotation and translation by minimizing the following objective function:

$$\begin{gathered}\{R^{calib}, T^{calib}\} = \arg\min_{R,T} \sum_{i=1}^{N} \|p_2^i - \frac{1}{s} c^{int}(Rp_3^i + T)\| \\ \text{subject to } R^T R = I_3, det(R) = 1\end{gathered} \tag{2}$$

where $p_2 \in \mathbb{R}^2$ and $p_3 \in \mathbb{R}^3$ are corresponding 2D and 3D keypoints. $R^{calib} \in \mathbb{R}^{3\times3}$ is the extrinsic rotation matrix, $T^{calib} \in \mathbb{R}^3$ is the extrinsic translation vector, and $s \in \mathbb{R}$ denotes the scaling factor [86, 89]. For the temporally smooth rotation and translation of a 3D pose across frames, we initialize the extrinsic parameter, R, and T, with the result from the previous frame. The 3D pose for each person, $p_3 \in \mathbb{R}^{3\times N}$, at each frame is calibrated (or localized) with the estimated corresponding extrinsic parameter.

$$p_3^{calib} = R^{calib} p_3 + T^{calib} \tag{3}$$

From the calibrated 3D poses, $p_3^{calib} \in \mathbb{R}^{3\times N}$, we remove people considered as the background. For example, for rope jumping competition scene, a set of people may rope jump while others are sitting and watching. Depending on the scene, not all people captured may partake in an activity (e.g., bystanders). To effectively collect 3D pose and motion that belongs to a target activity, we thus remove those people in the (estimated) background. We first calculate the pose variation across the frames as the summation of the variance of each joint location across time. Subsequently, we only keep those people with the pose variation larger than the median of all people.

2.3.2 Estimation of Camera Egomotion In an arbitrary video, the camera can move around the scene freely. However, the pipeline should not confuse the camera motion with human motion. For example, a person who does not move (much) may appear at a different location in subsequent frames due to the camera movement, which, however, is misleading for our purpose. Also, a moving person can always appear in the center of the frame, and thus erroneously appear static, if the camera follows that person and therefor the movements are effectively compensated for in the video. In these two cases, the virtual sensor should capture no motion (static), or the global body acceleration only, respectively, independently from camera motion. Hence, before generating the virtual sensor data, the 3D poses, which were previously calibrated per frame, need to be corrected for camera ego-motion, i.e., potential viewpoint changes, across the frames.

Camera ego-motion estimation from one viewpoint to another requires 3D point clouds of both scenes [6, 59, 67]. Deriving a 3D point cloud of a scene requires two pieces of information: *i)* the depth map; and *ii)* camera intrinsic parameters. For camera intrinsic parameter, we reuse the previously estimated parameters. The depth map is the distances of pixels in the 2D scene from a given camera center, which we estimate with the *DepthWild* model [22] for each frame. Once we have obtained the depth map and the camera intrinsic parameters, we can geometrically inverse the mapping of each pixel in the image to the 3D point cloud of the original 3D scene. With basic trigonometry, the point cloud can be derived from the depth map using the previously estimated camera intrinsic parameter, $c^{int} = [f_x, f_y, p_x, p_y, d]$. For a depth value Z at image position (x, y), the point cloud value, $[X, Y, Z]$, is:

$$[X, Y, Z] = \left[\frac{(x - p_x) \cdot Z}{f_x}, \frac{(y - p_y) \cdot Z}{f_y}, Z\right] \tag{4}$$

Once point clouds are calculated across frames, we can derive the camera ego-motion (rotation and translation) parameters between two consecutive frame point clouds. A popular method for registering groups of point clouds is the Iterative Closest Points (ICP) algorithm [6, 59, 67]. Fixing a point cloud as a reference, ICP iteratively finds closest point pairs between two point clouds and estimate rotation and translation for the other point cloud that minimizes the positional error between matched points [6]. Since we extract color point clouds from video frames, we adopted Park *et al.*'s variant of the ICP algorithm [55], which considers color matching between matched points in addition to the surface normals to enhance color consistency after registration. More specifically, we utilize background point clouds instead of the entire point cloud from a scene because the observational changes for the stationary background objects in the scene are more relevant to the camera movement. We consider humans in the scene as foreground objects, and remove points that belong to human bounding boxes from 2D pose detection. The reason for this step is that we noticed that including foreground objects, such as humans, leads to the ICP algorithm confusing movements of moving objects, i.e., the humans, and of the camera. With the background point clouds, we apply the color ICP algorithm [55] between point clouds at time $t - 1$ and t, q_{t-1} and q_t respectively. As such, we iteratively solve:

$$\{R_t^{ego}, T_t^{ego}\} = \arg\min_{R,T} \sum_{(q_{t-1}, q_t) \in \mathcal{K}} (1 - \delta)\|C_{q_{t-1}}(f(Rq_t + T)) - C(q_{t-1})\| + \delta\|(Rq_t + T - q_{t-1}) \cdot n_{q_{t-1}}\| \tag{5}$$

where $C(q)$ is the color of point q, n_q is the normal of point q. $\mathcal{K}$ is the correspondence set between q_{t-1} and q_t, and $R_t^{ego} \in \mathbb{R}^{3\times3}$ and $T_T^{ego} \in \mathbb{R}^3$ are fitted rotation and translation vectors in the current iteration. $\delta \in [0, 1]$ is the weight parameter for the balance between positional and color matches.

The estimated sequence of translation and rotation of a point cloud represents the resulting ego-motion of the camera. As the last step, we integrate the calibrated 3D pose and ego-motion across the video to fully track 3D human motion. Previously calibrated 3D pose sequences, p_3^{calib}, are rotated and translated according to their ego-motion at frame t:

$$p_{3_t}^{track} = R_t^{ego} p_{3_t}^{calib} + T_t^{ego} \tag{6}$$

where $p_3^{track} \in \mathbb{R}^{T\times N\times3}$ is the resulting 3D human pose and motion tracked in the scene for the video, and T is the number of frames, and N is the number of joint keypoints. The overall process of compensating camera ego-motion is illustrated in Figure 3.

2.4 Generating Virtual Sensor Data

Once full 3D motion information has been extracted for each person in a video, we can extract virtual IMU sensor streams from specific body locations. The estimated 3D motion only tracks the locations of joint keypoints, i.e., those dedicated joints that are part of the 3D skeleton as it has been determined by the pose estimation process. However, in order to track how a virtual IMU sensor that is attached to such joints rotates while the person is moving, we also need to track the orientation change of that local joint. This tracking needs to be done from the perspective of the body coordinates. The local joint orientation changes can be calculated through forward kinematics based from the hip, i.e., the body center, to each joint. We utilize state-of-the-art 3D animation software, i.e., Blender, to estimate and track these orientation changes. Using the orientation derived from forward kinematics, the acceleration of joint movements in the world coordinate system is then transformed into the local sensor coordinate system. The angular velocity of the sensor (gyroscope) is calculated by tracking the orientation changes of the sensor.

We employ our video processing pipeline on raw 2D videos that can readily be retrieved through, for example, querying public repositories such as YouTube, and subsequent curation (not within the focus of this paper). The pipeline produced virtual IMU, for example, tri-axial accelerometer data. This data effectively captures the recorded activities, yet the characteristics of the generated sensor data, such as MEMS noise, differ from real IMU data. In order to compensate for this mismatch, we employ the *Imusim* [83] model to that extracts realistic sensor behavior for each on-body location. *Imusim* estimates sensor output considering mechanical and electronic components in the device, as well as the changes of a simulated magnetic field in the environment. As such, this post-processing step through *Imusim* leads to more realistic IMU data [4, 37, 57].

2.5 Distribution Mapping for Virtual Sensor Data

Although the extracted sensor stream may capture the core temporal patterns of the target activity in the estimated 3D motion, the intrinsic characteristics of the virtual sensor can be far from that of the actual physical sensor used for the activity recognition. As the last step before using a virtual sensor dataset for HAR model training, we transfer the distribution of the virtual sensor to that of the target sensor. For computational efficiency, the rank transformation approach [15] is utilized:

$$x_r = G^{-1}(F(X \leq x_v)) \tag{7}$$

where, $G(X \leq x_r) = \int_{-\inf}^{x_r} g(x)dx$ and $F(X \leq x_v) = \int_{-\infty}^{x_v} f(x)dx$ are cumulative density functions for real, x_r, and virtual, x_v, sensor samples, respectively. In our experiments, we will show that only a few seconds to minutes of real sensor data is sufficient to transfer the virtual sensor effectively for successful activity recognition.

3 Training Activity Recognition Classifiers with Virtual IMU Data

We now describe a series of experiments to examine the viability of using IMUTube to produce virtual IMU data useful for HAR. We first consider the performance of virtual IMU data on a HAR dataset, which provides both video and real IMU data to enable a fair comparison between virtual and real IMU data. Here, we see promising results, which suggests that training activity classifiers from virtual IMU data alone can perform well on real IMU data. We then move on to show that activity classifiers trained using this virtual IMU data can also perform well on real IMU data coming from common HAR datasets, namely Opportunity [13] and PAMAP2 [64]. Finally, we describe how we curate a video dataset comprising of online videos (e.g., YouTube) in order to extract virtual IMU data for complex activities. In each experiment we compare the performances of models on real IMU data (i.e., the test data is from real IMUs), when trained from real IMUs (R2R), trained from virtual IMUs (V2R), or trained from a mixture of virtual and real (Mix2R) IMU data.

3.1 Feasibility Experiment under Controlled Conditions

There are many sources of potential noise in the IMUTube pipeline that may impact building a model of activity. Therefore, in our first experiment we hold constant as many factors as possible. We accomplish this by using the RealWorld dataset [76], an activity recognition dataset that contains not only IMU data but also provides videos of the subjects performing the activities.

Data The Realworld dataset covers 15 subjects performing eight locomotion-style activities, namely *climbing up, climbing down, jumping, lying, running, sitting, standing*, and *walking*. Each subject wears the sensors for approximately ten minutes for each activity except for jumping (<2 minutes). The video and accelerometer data are not time-synchronized, as each video starts some time (under one minute) before each activity begins. The video is recorded using a hand-held device by the experiment's adminstrator, who follows the subject as they perform the activity (e.g. running through the city alongside the subject). The videos do not always present a full-body view of the subject, and the video-taker sometimes makes arbitrary changes to the video scene (e.g., he/she might walk past the subject, or rotate the camera from landscape to portait mode half way). These factors present extra difficulty in extracting virtual IMU for the full duration of the activities; nonetheless we are able to extract 12 hours of virtual IMU data, this is compared to 20 hours of available real IMU data. As a preprocessing step, we remove the first ten seconds of each video and divide the remainder into two minute chunks for efficient running of IMUTube. We assume all IMU data to have a frequency of 30Hz. We use sliding windows to generate training samples of duration 1 second and 50% overlap. The resulting real and virtual IMU dataset contains $221k$ and $86k$ windows, respectively. In R2R, we use IMU data from subject 14 for validation, subject 15 for test and the rest for training. In V2R, we follow the same scheme except we use virtual data from subject 1 to 13 for training. In Mix2R, we follow the same scheme but use both real and virtual IMU data.

Method We employ two machine learning models, Random Forest and DeepConvLSTM [54]. Random forest is trained using ECDF features [25] with 15 components, and DeepConvLSTM is trained on raw data. For DeepConvLSTM, we train the model for a maximum of 100 epochs with the Adam optimizer [38], early stopping on the validation set with a patience of ten epochs. We follow standard hyperparameter tuning procedures using grid search on a held-out validation set; learning rate is searched from 10^{-6} to 10^{-3}, and weight decay is searched from 10^{-4} to 10^{-3}. To further regularize the model training, we additionally employ augmentation techniques from [80] with a probability of application set at either 0 and 0.5 depending on validation set result.

We evaluate classification performance using mean F1 score with Wilson score interval (95% confidence). All reported F1 scores are an average of three runs initiated with a different random seed. We reuse these settings throughout the paper unless otherwise specified. In both cases of DeepConvLSTM and Random Forest, we report the highest test F1-score achieved using varying amounts of training data. We discuss this effect of train data sizes in Section 4.3.

Results In Table 1, we see convincing evidence that human activity classifiers can learn from virtual IMU data. When training purely from virtual IMU data (V2R), the models can recover on average 80% of the R2R performance. For example, using a Random Forest we see an F1-score of 0.63 from virtual IMU alone, in comparison training using real IMU data achieves 0.74. We observe part of the reason for this lower performance is that the accuracy for models from virtual IMU data saturates very quickly – after just 1,000 windows per class (Section 4.3).

Furthermore, when we train from a mixture of real and virtual IMU data (Mix2R), we even observe on average a performance *gain* of 6.0% beyond that achieved by training from real IMU data alone. This result further suggests that, not only can virtual data be used to train HAR classifiers, they can be added to real data to improve performance.

We find that the V2R setup poses more challenges to learning the DeepConvLSTM. We hypothesize this is because DeepConvLSTM learns from raw data, whereas the Random Forest classifier uses ECDF features. As a

Table 1. Recognition results on the Realworld dataset (8 classes) when using different training sets. Wilson score confidence intervals are shown. Models trained solely on virtual IMU data (V2R) recover 80% of the R2R model performance on average. Models trained with mixed data (Mix2R) additionally gained 6.0% mean F1 score (on average) and surpassed R2R model.

Model	R2R	V2R	Mix2R
Random Forest	0.7401±0.0111	0.6321±0.122	0.7797±0.0105
DeepConvLSTM	0.7305±0.0073	0.5465±0.0082	0.7785±0.0068

consequence, the DeepConvLSTM may be learning highly specific features related to virtual IMU data which struggles to immediately generalize to the real IMU data. Though the DeepConvLSTM result is not as strong as Random Forest, we still see V2R for DeepConvLSTM matches 75% of its R2R performance. We also see that this issue is resolved when using a mix of virtual and real data for training, and Mix2R even outperforms R2R by 6.6%.

In these results, we have so far only considered relatively straightforward techniques in extracting and modelling the virtual IMU data. We delve into these concerns in Section 4 to provide a more complete view.

3.2 Performance on Common Activity Recognition Datasets

We have achieved promising results under the controlled conditions of Realworld, which simultaneously gathers video and IMU together. We now seek to relax these conditions, and establish the viability of IMUTube when the exact action performed in the video data and the testing real IMU data do not completely align. Imagine a scenario where we want to build a classifier for standing vs sitting. Instead of collecting simultaneous video and real IMU data of people standing and sitting, we want to leverage existing videos of people standing and sitting and train the classifier using the derived virtual data. In the following, we test this scenario by re-using the video data from Realworld and using its virtual data to test on two common HAR datasets, Opportunity and PAMAP2. These datasets are considered as they contain activity labels that roughly correspond to those in Realworld.

Data We consider activities in Opportunity and PAMAP2 which are overlapping with those in Realworld, i.e., four classes (*stand, walk, sit, lie*) in Opportunity, and eight classes (*ascending stairs, descending stairs, rope jumping, lying, running, sitting, standing*) in PAMAP2. In both cases, we use 1-second sliding windows with 50% overlap.

For Opportunity, we re-extracted virtual data from the Realworld videos in eleven body positions (left and right feet, left shin and thigh, hip, back, left and right arms, left and right forearms), which resulted in $40k$ and $46k$ real and virtual IMU windows respectively. We use the data from the last subject (4) as test set.

For PAMAP2, the activities are slightly different from those in Realworld so we equated the labels with the closest meaning (e.g., using *jumping* Realworld videos as the source for *rope jumping* virtual IMU in PAMAP2. The PAMAP2 dataset specifies that sensors were placed in three locations (dominant wrist, dominant ankle, chest), which gives rise to a total of four possible combinations for arm and chest location when we extract virtual IMU data from a single video (i.e., left-left, right-right, left-right, right-left). We took advantage of this ambiguity and extracted 4× as much virtual IMU per video, resulting in $24k$ and $152k$ windows for real and virtual IMU respectively. We use data from subject 6 as test set (same as [26]).

Results Table 2 shows the classification performance for R2R, V2R and Mix2R. For Opportunity, where we have a similar amount of available real and virtual IMU data, the models are able to achieve an average of 91% of R2R performance when training from virtual IMU data alone. This is an encouraging result as the V2R/R2R ratio is even higher than in Realworld (80%) where conditions are held constant. While this good performance might be related to the simplicity of the motions classified (four locomotion actions),the conditions of data collection in Realworld and Opportunity are very different–the subject could be walking through the forest or city in Realworld, but all subjects perform activities inside a laboratory in Opportunity–so being able to utilize virtual

Table 2. Activity recognition results (mean F1 score) on locomotion activites found in PAMAP2 (8 classes), and Opportunity dataset (4 classes) when using different training data. Across datasets and models, V2R model recovers approximately 89% of the R2R model performance. When real IMU data is mixed for training (Mix2R), the model was comparable to R2R model (approximately, 99.9% of R2R F1 score on average).

Dataset	Model	R2R	V2R	Mix2R
Opportunity (4-class)	Random Forest	0.9122±0.0056	0.8493±0.0070	0.9150±0.0055
	DeepConvLSTM	0.8871±0.0074	0.7882±0.0096	0.8838±0.0075
PAMAP2 (8-class)	Random Forest	0.7386±0.0152	0.7030±0.0158	0.7350±0.0152
	DeepConvLSTM	0.7002±0.0161	0.5524±0.0175	0.7020±0.0161

data in one scenario and test on the other is not a straightforward task. For PAMAP2 (8-class), the models are able to achieve an average of 82% of R2R performance when training from virtual IMU data alone. Again, the conditions of data collection in Realworld and PAMAP2 can be quite different–the subject could be climbing down the streets of a city (a mixture of pavement and stairs) in Realworld whereas all subjects are climbing up the same building in PAMAP2. There is also a slight shift of the meaning of activity labels in this case. These results suggest that, on these two tasks, virtual IMU data is capable of capturing salient features that are generalizable and robust across testing scenarios.

As the data generation conditions between real and virtual IMU data are quite different, we did not expect an activity classifier trained from a mixture of two types of data to bring performance gains over one which was trained from real IMU data only. This is indeed the case as we observe no significant increase in F1 score when comparing Mix2R against R2R on both tasks. Nonetheless, this set of results overall presents an encouraging view which aligns well with our vision for IMUTube–that virtual IMU data, even when collected under vastly different settings, can be useful in building capable classifiers that work well on real IMU data.

3.3 Virtual IMU Data for Complex Activity Recognition

Encouraged by the results so far, we now try to apply IMUTube onto activity recognition scenarios with even more challenging conditions and test its ability in building classifiers for complex activities. Our ultimate vision for IMUTube is to extract virtual data from any video, especially those freely available in large online repositories such as YouTube. To test the feasibility of doing so, we first need to curate a dataset with activity videos originated from the web. In the following, we attempted to source these videos for complex activities present in PAMAP2, and trained activity classifiers from the extracted virtual IMU data.

Data We curated a dataset of virtual data covering four complex activities present in PAMAP2, namely *vacuum cleaning, ironing, rope jumping* and *cycling*. To efficiently locate such videos, we extract annotated video segments from activity video datasets in the computer vision domain, including ActivityNet [9], Kinetics700 [12], HMDB51 [39], MPIIHPD [3], UCF101 [74], Charades [73], AVA [23], MSRdailyactivity3D [43], and NTU RGB-D [45]. The resulting video dataset consists of a mix of videos collected in experiment scenarios and in-the-wild (e.g., from YouTube). In total, we collected ∼ 10 hours of virtual data from 7,255 videos. To extend our activity recognition task to as many classes in PAMAP2 as possible, we also reuse the other seven videos from Realworld (we do not use the *jumping* videos); this allows us to consider an 11-class activity recognition problem in PAMAP2. As mentioned for the PAMAP2 (8-class) task, we face an ambiguity in sensor location which led us to extract 4× virtual data per video. Using sliding windows of 1-second size and 50% overlap, we resulted in $38k$ real and $390k$ virtual IMU windows in total.

Results For these challenging conditions (using in-the-wild videos, learning complex activities), Table 3 shows that virtual IMU data can still be useful for training activity classifiers. On average, training from virtual data

Table 3. Recognition results (mean F1) on PAMAP2 (11-classes) when using different training data. The model trained with virtual IMU data including YouTube videos (for four complex activities) recovered 80% of R2R model performance on average. For Mix2R model, additional real IMU data helped models increase performance up to 97% of R2R model performace.

Model	R2R	V2R	Mix2R
Random Forest	0.7724±0.0116	0.6508±0.0132	0.7077±0.0126
DeepConvLSTM	0.6977±0.0129	0.5326±0.0140	0.7095±0.0128

alone recovers 80% of the R2R performance. We acknowledge that DeepConvLSTM achieves 0.53 F1 score under V2R, a weak result in comparison to real IMU performance (0.69 F1 score). We believe this is due to a combination of domain shift between the datasets, as well as limitations in quality of data IMUTube is currently able to extract – that are exacerbated by complex activities introduced in this experiment. Section 4.2 highlights significant domain shifts that results in loss in accuracy, even when only real IMU data is involved.

As expected in the Mix2R case, we see no signs of benefit in mixing virtual and real IMU to train the model, as Random Forest has a 8% decrease in F1 score than the R2R counter part. This is again related to the domain shift which exists within the training data, as the data came from different sources and interpretations of activity labels. To better cope with the scneario where we want to make use of both real and virtual IMU data, we need to understand the domain shift better and investigate more sophisticated methods, which we will in Section 4.2 and Section 5, respectively. DeepConvLSTM, however, does not suffer from such drop in performance when mixing real and virtual data. We presume that the difference comes from the feature learning capabilities of deep learning models, which learn better when more data is available. We also examine the quality of the generated virtual IMU in Section 4.1, with directions to improve virtual data quality given in Section 6.2.

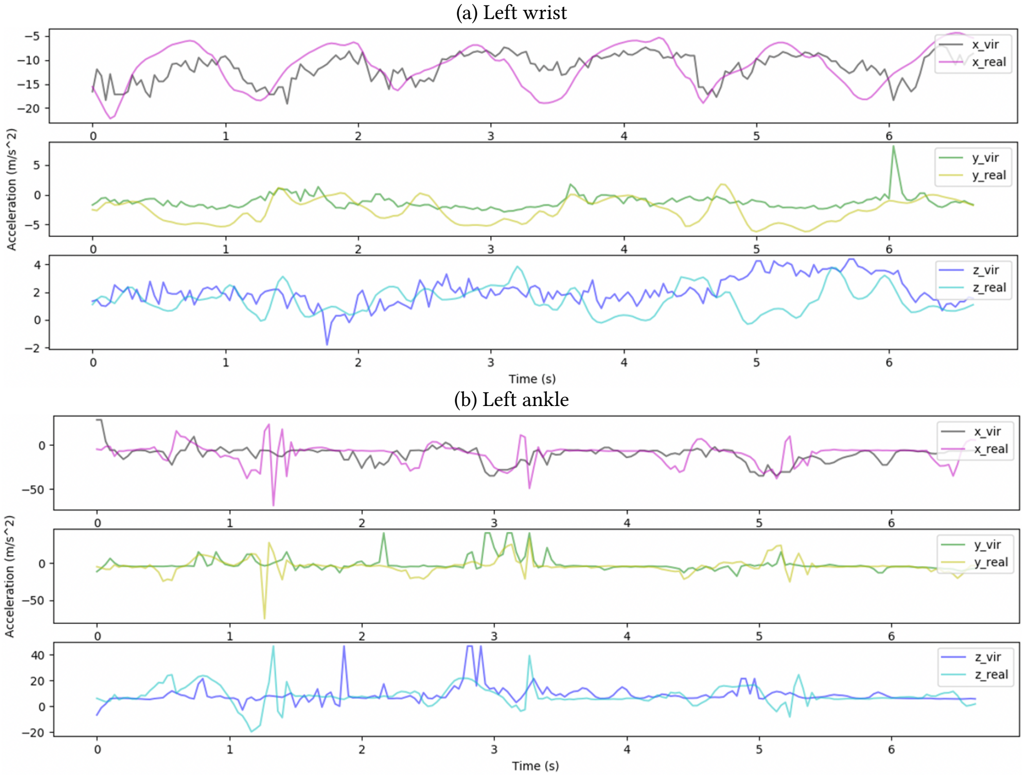
Over this set of experiments, we have established a proof of concept that the virtual IMU data generated from our pipeline can accurately represent the motion aspect of a range of locomotion and more complex activities, to the extent that models can be trained purely from virtual data and perform reasonably well on real IMU data.

4 Understanding Virtual IMU Data

Across multiple datasets, the model trained on virtual IMU dataset (V2R) performed well for the real IMU test datasets. However, the V2R model performed between 74% - 93% compared to R2R model which was trained on real IMU dataset, and only could equal or outperform the R2R when trained with real IMU data (Mix2R). Particularly for the experiment on PAMAP2 with eleven classes, the V2R model could not outperform R2R even with the larger training set from YouTube compared to R2R model. Thus, in this section, we investigate the potential source of this performance gap through analyzing the generated virtual IMU data in detail. First, we compare accelerometer sample level similarity between the synced sequence of virtual IMU and related IMU data. Then, at a distribution level, the effect of domain shift between the virtual IMU and real IMU is investigated, along with the impact of distribution mapping post processing on the virtual IMU data (as described in Section 2.5). Finally, we conduct further analysis on mixing virtual and real IMU for model training (Mix2R) as the Mix2R showed comparable, or even superior performance, relative to R2R model. In particular, we evaluate the influence of the different virtual IMU training data sizes for Mix2R. Through the analysis, we aim to provide key insights into the IMUTube pipeline and the use of virtual data for human activity recognition.

4.1 Comparing Virtual and Real IMU

We do not expect IMUTube, especially in this first proof-of-concept design, to function flawlessly. Given the complexity of the process, the translation from video to virtual IMU data will naturally still contain errors. Despite this, we observed promising results of competitive V2R performance in the prior section. This seems


(a) Left wrist

(b) Left ankle

Fig. 4. Comparison between virtual and real IMU on the TotalCapture dataset. Distribution mapping has been applied for the virtual IMU data.

to suggest that perfect sample-level realism in the virtual IMU data is not necessary to train capable human activity classifiers. In the following, we compare virtual and real IMU samples to better understand the limits of IMUTube currently – and argue that the focus, during virtual IMU generation, should be placed on capturing salient features useful for activity recognition.

Method Sample-level comparison between the virtual and real IMU data requires a dataset with time-synchronized video and IMU sensor. Although Realworld dataset contains both accelerometer and video data, these two modalities are not synchronized (as mentioned in Section 3.1). Therefore in this experiment, we introduce the TotalCapture dataset [79], as it contains the time-synchronized (real) IMU data and video which virtual IMU data is extracted. The TotalCapture dataset was created to capture humans performing various scripted motions, and so does not contain labels that are immediately useful for activity recognition-related tasks. For this reason, we did not use it in Section 3.

Analysis Figure 4 shows an example of the virtual and Real IMU time series of a person walking, with sensors placed on his/her wrist and ankle. Along the x-axis, virtual IMU readings are seen to reflect large movement changes as in the real IMU—one can almost see from the 'wrist' time series (see Figure 4(a)) that the person is walking with periodic hand movements. Along the z-axis, the virtual IMU is also seen to capture any spikes in acceleration reasonably well, albeit with a noticeable time lag in the 'ankle' case. Virtual and real IMU data differ

Table 4. Recognition results (mean F1) on Opportunity (4 classes) when using training data from different datasets. There is a significant drop in performance when using train data collected under different circumstances than the test case.

Train data source	Without mapping	With mapping
Virtual data	0.1961±0.0035	0.8493±0.0029
PAMAP2	0.2770±0.0040	0.8275±0.0034
Realworld	0.2828±0.0041	0.8084±0.0036
Opportunity	0.8823±0.0029	-

the most along the y-axis. We postulate that this is related to a dimensionality issue—we are trying to reconstruct 3-D information from a 2-D image time series. The y-axis here refers to the axis pointing perpendicular to the vision pane, which means any acceleration measured along this dimension cannot be easily deduced visually.

While generating realistic virtual IMU data is important, this is secondary to our main goal of producing virtual IMU data that captures useful information for HAR tasks. To achieve this, what is vital is the ability of virtual IMU data to capture salient features of the activities that we need to recognize. We already see signs of this happening with the current IMUTube (e.g., the x-axis of the 'arm' while walking in Figure 4). This also offers a possible explanation for the better V2R performance seen in predicting locomotion-style activities in Section 3. Perhaps IMUTube, in its current form, is best suited to capture information about simple motions (i.e., ones mostly characterized by movement in a 2D plane) of which there are still a wide variety, and to which existing HAR methods still struggle to generalize [40]. To apply IMUTube to more complex activities it may require improved techniques during virtual IMU data generation – see Section 6.2.

4.2 Coping with Domain Shift

The last step of our pipeline performs a distribution mapping post-processing (Section 2.5) between virtual and real data. Applying some form of distribution mapping is necessary due to the presence of a *domain shift* between the training and testing data. This domain shift is not exclusive to extracting virtual sensor data from videos, but it is also present whenever data is taken from different tasks (or datasets) which result in dissimilar data distributions between training and testing.

Method To illustrate the effect of such domain shift, we took training data from different sources (including virtual IMU data, PAMAP2, and Realworld) and trained a Random Forest to test on the Opportunity 4-class locomotion classification task. We used virtual IMU data (approximately 90 minutes per activity) extracted from Realworld videos for four locomotion classes in Opportunity.

Analysis In Table 4, we observe a significant drop in the performance when a Random Forest is trained using virtual IMU data, PAMAP2 or Realworld without any distribution mapping. Having applied the same distribution mapping technique from IMUTube to all sources, we even observe that training from virtual data outperforms training from other real IMU datasets. This hints that virtual data might have greater value than current IMU data into developing general HAR models. We arrive at the conclusion when the same analysis was conducted on PAMAP2 and Realworld datasets. We emphasize that the distribution mapping is performed on virtual IMU data only with the subset of the real IMU training data from Opportunity dataset. We also evaluated the final V2R performance according to the varying amount of real IMU training data used for distribution mapping, and further observed that only small subset (1̃0 minutes per class) of real IMU data for distribution mapping is sufficient to guarantee the reasonable V2R performance.

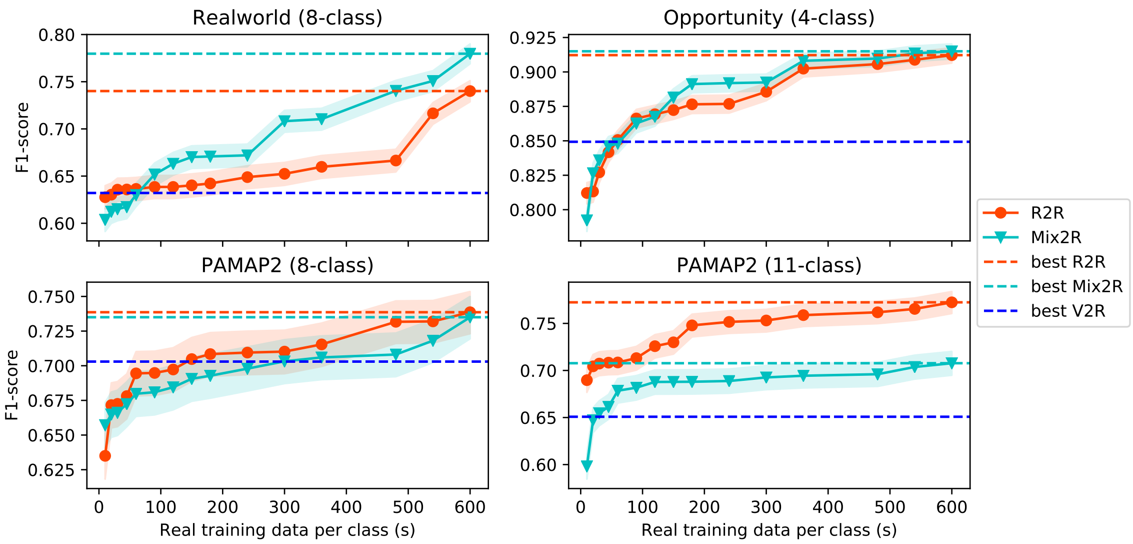


Fig. 5. Mix2R and R2R performance of a random forest model on 4 different HAR tasks when different amounts of real data per class (in seconds) is available for training. Note that the ratio of virtual data and real data is kept at 1:1 at all datapoints.

4.3 Varying the Mixture and Size of Training Data

In what follows, we inspect how varying the mixture and size of training data affects recognition performance.

Method To examine how learning performance changes as the mixture and size of training data is varied, we performed two investigations. First, we reported on each dataset how R2R and Mix2R performance on Random Forest compares with different amounts of real training data, while fixing the ratio of real-to-virtual IMU at 1:1. Second, we looked more closely at case of using DeepConvLSTM for the Realworld Data (the case where Mix2R outperforms R2R the most) and reported the result where the ratio of real-to-virtual IMU is varied.

Analysis In Figure 5, the learning curve for the different amount of real IMU training data is shown. Mix2R performance approximates R2R with 10 minutes of real data added per class in 3 out of 4 datasets. In Realworld, where the settings of virtual and real IMU data are the most similar, it is clear that the learning rate of Mix2R is consistently higher than R2R. The learning curves follow similar trajectories in Opportunity and PAMAP2 (8-class). However, in PAMAP2 (11-class), the real and virtual IMU learning curves diverge when we add 100 seconds of real IMU data per class, and both plateau soon after. Given that PAMAP2 (11-class) is also the case where we predict complex activities under the most dissimilar settings, its plot highlights the difficulty of the classification task for both real and virtual IMU data. Although it may appear that learning accuracy plateaus with relatively small amount of data per class (600 seconds per class, for instance) – we highlight that this has been commonly observed in the literature for the HAR datasets we used(Opportunity, PAMAP2, e.g., [28, 65, 87]).

In Section 3.1, we saw that the best Mix2R performance demonstrated by DeepConvLSTM on the Realworld dataset, when the real-to-virtual ratio was at 1:1. Here, we are interested in determining how the performance might vary when increasing quantities of virtual IMU training samples are added. Table 5 considers the Realworld dataset, and demonstrates that the model performance changes as the ratio of virtual IMU data to real IMU data is increased, while keeping the number of real training windows per class at 300. The performance peaks when equal amounts of real and virtual data are available (9.7%), and decreases as more virtual IMU data is added to the

Table 5. Activity recognition results (mean F1 score) on the Realworld dataset (8 classes) Different amounts of virtual data is added to a constant amount of real data (300 windows).

No. of real windows	No. of virtual windows	Real : Virtual	F1-score
300	0	1:0	0.7231±0.0126
300	300	1:1	0.7931±0.0114
300	600	1:2	0.6856±0.0131
300	1500	1:5	0.7302±0.0125
300	3000	1:10	0.7282±0.0125

training set, dipping by 5.2% when their ratio is at 1:2. This shows that the effect of mixing virtual and real data is not straightforward. It is possible that as more virtual IMU data is used, the domain shift issue becomes severe and the DeepConvLSTM starts to overfit to the virtual IMU data; and at the same time, the diversity in virtual IMU data also increases as more virtual windows are added, so V2R performance picks up again.

Hence, we suggest finding the right balance between the amount of real and virtual IMU data for a model to learn the target activity pattern coexisting in both real and virtual IMU data, before overfitting to the virtual IMU data. We also anticipate that as the quality of virtual IMU improves in future versions of IMUTube, that larger amounts of it will be able to be successfully integrated during HAR training.

5 Transfer Learning with Virtual IMU Data for HAR Classifiers

The previous two sections demonstrate that sensor based human activity classifiers can learn from virtual IMU data, although limitations still exist. So far, we assume that labeled virtual and real IMU datasets for target activities are always available. We explored direct supervision with mixing of the real and virtual IMU data. In practice, such a scenario may not always be feasible. Curating video datasets for virtual IMU data could be challenging, as titles or description of videos can be arbitrarily ambiguous. Thus, virtual IMU data for some target activities may not be immediately ready for direct application.

Here, we explore two additional cases for utilizing the virtual IMU data: *i)* when the virtual IMU dataset contains a subset of the the target activities; and, *ii)* when labels for the virtual IMU dataset are not available at all. To study these cases, we leverage transfer learning. First, we investigate a supervised transfer learning approach, i.e., a model pre-trained is on the virtual IMU data, and subsequently fine-tuned on the real IMU data. Here, the labels used for pre-training and fine-tuning need not match. Next, we explore unsupervised transfer learning, i.e., a model is pre-trained on unlabelled virtual IMU data to learn useful feature representations from sensor readings. The learned weights are used to extract features for the real IMU data. These representations are used to train a classifier. Considering these approaches extends the usefulness of IMUTube towards handling realistic issues in label collection, as we do not yet incorporate any automated video labelling or search mechanisms. Notably, we are able to achieve a significant improvement of 10% on the Realworld test dataset, through supervised pre-training on the virtual IMU data, over a model trained from scratch. We also show the viability of using unsupervised transfer learning, which demonstrates competitive performance on the same task. Table 6 summarizes the results.

5.1 Supervised Transfer Learning

We investigate a transfer learning setup which uses both labelled virtual and real IMU data. Here, we explore two scenarios for using supervised transfer learning. In the first set of experiments, we consider the simplest scenario where the base (virtual) model and fine-tuned (real) model are learnt on the same activity recognition task. To illustrate the idea, let's assume that we already have curated video data for the target activities. Then, only a small amount of real IMU data will be collected until the virtual IMU dataset is extracted from all the videos. In

Table 6. Recognition results (mean F1) of transfer learning setups when evaluated on different HAR tasks. R2R is the baseline trained on real data from scratch. Transfer learning (TL) results shows the performance of the models finetuned on real data.

	DeepConvLSTM		CAE+RF	
	Supervised R2R	Supervised TL	Unsupervised R2R	Unsupervised TL
Realworld	0.7305±0.0073	0.8337±0.0061	0.7923±0.0067	0.7718±0.0069
Opportunity	0.8871±0.0074	0.9100±0.0067	0.8896±0.0074	0.8477±0.0084
PAMAP2 (8-class)	0.7002±0.0161	0.7137±0.0159	0.6471±0.0168	0.6809±0.0164
PAMAP2 (11-class)	0.6977±0.0129	0.7023±0.0129	0.7004±0.0129	0.6989±0.0129

such a case, instead of waiting until the sufficient amount of real IMU data is collected, we can first train the model on the virtual IMU data and fine-tune on the small scale real IMU data.

Next, we consider a scenario where the virtual IMU data only contains a subset of the real IMU data activity classes. We investigate how a model trained on a large-scale dataset with limited activity classes can adapt to the small-scale target dataset, which contains a higher number of classes. To examine this, we pretrain a model on the virtual PAMAP locomotion (8-classes) task and fine-tune it on the real PAMAP locomotion (8-classes) and complex activities (11-classes) tasks.

Method Our supervised transfer learning protocol consists of two stages: pre-training a model on the virtual IMU data, followed by fine-tuning with the real IMU data. We utilize the DeepConvLSTM architecture for the supervised transfer learning experiments. In the first experiment, we pretrain the network on the corresponding virtual IMU data for each real IMU dataset (using a train/validate/test split of 80%/10%/10%). We choose the pre-trained model with the highest target real IMU validation set performance for fine-tuning with the real IMU train dataset. During fine-tuning, all model weights are updated and we report the performance on the real IMU test dataset. For the second experiment, we pre-train the model using the same protocol as the previous experiment. While fine-tuning, we replace the last layer of the pre-trained model with the target number of classes (thereby going from 8 to 11 activity classes for PAMAP2) and update all the network weights.

Results When the virtual and real IMU data contain the same labels, pre-training improves the performance over training from scratch on the real IMU data (Table 6). For all datasets, we obtain statistically significant performance gains by pre-training, with Realworld exhibiting the most substantial gain of 6.7%. Figure 6 shows that only a small amount of real data is needed to fine-tune the base model, such that it surpasses R2R performance.

Supervised transfer learning is also effective when the virtual IMU data contains only a subset of class labels from the real IMU data. Through transfer learning, the model achieves an F1-score of 0.71, which demonstrates an statistically significant improvement of 1.35% over R2R trained from scratch. Although there is a label mismatch between the virtual and real IMU data, the model benefits from the large scale of the virtual IMU data.

5.2 Unsupervised Transfer Learning

Unsupervised transfer learning considers the scenario where we extract the virtual IMU data from a large body of videos without labels. Curating a collection of unlabeled videos is easier relative to obtaining labeled videos, particularly in scenarios where the video descriptions/labels maybe unreliable. Without a set of specific target activities in mind, any videos with human can be utilized. Hence, it allows us to curate a large collection of virtual IMU data consisting of very diverse movements and activities from which a model can learn generic representations.

Method The unsupervised transfer learning consists of two stages: the first involves pretraining a convolutional autoencoder (CAE) on the virtual data, while the second consists of fine-tuning on the real data. We use

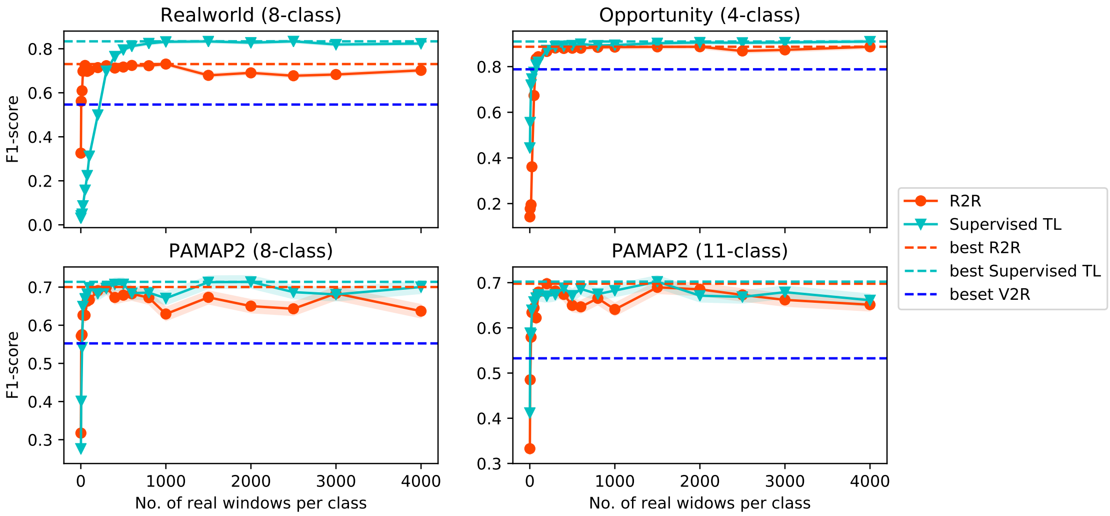


Fig. 6. Transfer learning vs. amount of real data used for training.

Haresamudram et al.'s architecture, where the encoder contains four convolutional blocks, leading to the bottleneck layer [28]. Each block contains two 3 x 3 convolutional layers followed by 2 x 2 max-pooling. Batch normalization is applied after each layer [32]. The output from the last convolutional block is flattened before being connected to the bottleneck layer. The decoder inverts the encoder by performing convolution, interpolation, and padding, in order to match the sizes of the corresponding encoder blocks [51]. ReLU activation [50] is used throughout, except the output, where the hyperbolic tangent function is used instead.

In the second stage, the representations for the real data are obtained by performing a forward pass on the pretrained encoder from the CAE. We compute the representations on the entire real IMU dataset. The representations from the real IMU training data are used to train a random forest classifier. Further, we also compare the effect of utilizing virtual IMU data for pretraining against using real IMU data.

Results In Table 6, we observe that virtual IMU for pre-training results in comparable performance to models trained on real IMU data, and on PAMAP2 (8-class), produce an improvement of 3.5% over the R2R protocol. Although we do not see evidence of this unsupervised TL scheme bringing any performance gains compared to R2R in this set of results, they demonstrate the feasibility of utilizing such schemes in scenarios where video labels are completely absent.

6 Discussion

In this section, we discuss the implications of the results presented, limitations in our approach, and highlight opportunities which this work opens up.

6.1 Establishing a Proof of Concept

We have presented a processing pipeline and a series of validation studies to support our thesis that an automated pipeline from video to virtual IMU data can replace the labor-intensive practice of collecting labelled datasets from real on-body IMU devices. The automated pipeline raises the opportunity to collect much larger labelled data sets, which in turn can improve classifiers for human activity recognition.

While our proof of concept system, IMUTube, does support the overall thesis (i.e., virtual IMU data automatically generated from videos both reduces effort to collect labelled datasets and improves HAR results), there are clear steps forward to advance this line of research.

Our validation experiments explored ways to best leverage data from both labelled virtual and real data, in situations where they are both available. Under a controlled setting (as in Realworld), a simple mixing of data from the two sources is seen to bring significant benefit to the activity recognition. Yet, when we consider scenarios where activity labels are more loosely defined and settings less controlled for (as in PAMAP2 11-class), mixing actually hurt performance. To address this shortcoming, we investigated transfer learning techniques, which presented performance gains on all tasks considered, except again in the case of PAMAP2 11-class.

PAMAP2 11-class is a special case as the activity recognition task extends to complex, non-locomotion human activities, such as vaccum cleaning. It is also different because we have utilized a diverse range of video data collected over multiple visual datasets. Our results show that we can indeed still learn from virtual data under such settings, and our V2R results still come to at least 76% when compared to R2R (Table 3) . However, what is still missing is that we have not seen significant improvement over R2R results through mixing (1.7% at best) or transfer learning (0.66% improvement). While this suggests that modelling complex activities and mixed data sources remain an issue, we believe that modelling complex activities, and by extension, the merit of using more accelerometery data (be it virtual or real) still warrants further investigation. For example, is there simply an upper bound to predicting these complex activities using motion-based data alone?

6.2 Limitations and extensions of current approach

We have utilized a series of off-the-shelf techniques at every step of the proposed pipeline in Section 2. While this supports reproduciblity of our results, it does result in limitations that impact overall quality of labelled data for HAR. We discuss some of the known limitations here at each step of the pipeline.

6.2.1 From Vision To Pose Accurate recovery of the human skeleton pose from videos has known limitations arising from the movement of both the subjects as well as the camera. For the former, we would expect improvement based on solutions that leverage more sophisticated pose tracking techniques that are more robust to vigorous movement, a change of scenery, the presence of multiple people, and occlusion. For the latter, camera movement relative to the people depicted in the videos could come from the instability of camera (e.g., hand-held cameras) as well as video filming techniques (e.g., panning shots). Specialized video stabilization strategies or camera ego-motion techniques can address these issues [71, 84, 88]. We believe that the application of these techniques (and others not yet mentioned or even developed) will further improve pose extraction quality and expand the variety of videos which can be treated as input to our pipeline.

6.2.2 From Pose To Accelerometry Our current approach assumes an equivalence between acceleration measured by a device on the wearer's body with that measured at the nearest body joint. This view discounts any consideration of factors such as body mass, device movement and skin friction. To better model the on-body location of IMU devices, utilizing techniques from body mesh modelling is a straightforward solution to increase realism to the pipeline. We foresee that investigating the use of body mesh might also bring up the possibilities of synthesizing credible accelerometry data from people of different body shapes from the movement of a single human skeleton pose [34, 46, 61]. In addition, while we have only considered the generation of virtual accelerometry data in this work, we can adapt most parts of the pipeline to generate the full set of IMU signals, including gyroscope and magnetometer readings.

6.2.3 From Accelerometry To virtual IMU Real IMU data, which have been the basis of building HAR classifiers, are not free of noise. Sensor noise may come from factors such as drift, hystersis and device calibration. To carry over such characteristic sensor noise on our virtual data, domain adaptation techniques can be deployed as well as more sophisticated techniques like Generative Adversarial Networks. [21, 66]

6.2.4 Learning from virtual data A domain shift exists when a machine learning model is trained from virtual data and tested on real IMU data. Again, domain adaptation strategies to the input of the machine learning model is a solution. Alternatively, it will be promising to investigate domain-invariant features learnt from virtual and real data, which could potentially lead to performance gains in HAR.

6.3 The Road Ahead

Our primary goal in this paper is to motivate the HAR community with a promising approach that overcomes the main impediment to progress—lacking large labelled data sets of IMU data. While technical challenges remain, we have validated this approach and provide a processing pipeline that the community can collectively develop. Here we highlight the most compelling research opportunities.

6.3.1 Large-scale data collection The ultimate goal, as suggested by the name of IMUTube for our initial tool, is to develop a fully automated pipeline that begins with the retrieval of videos representing particular human activities from readily available sources (e.g., YouTube) and converts that video data to labelled IMU data. Since it is much more common to have video evidence of the wide variety of human behaviors, this is an obvious advantage over past labor-intensive and small-scale efforts to produce such HAR datasets. We have shown great promise with this direction, and above listed some known limitations that can be addressed by different vision, signal processing, and machine learning techniques. The reader will note that the videos used for our validation studies were also curated, meaning there was significant effort in selecting appropriate video examples. The hope is that this curation effort can also be reduced and ultimately eliminated because the sheer number of relevant videos will overcome the deficiencies of less useful video data.

6.3.2 Deep learning Deep learning has transformed recognition rates in other fields [29, 53], but HAR has lagged, again due to the lack of large corpora of labelled data. While we expect that IMUTube is a significant advance towards that goal, having the data alone is not the end goal. We have not yet produced a large-scale HAR dataset, and until we do so we can only hope that deep learning techniques will take over. We then fully expect HAR to inform deep learning techniques.

6.3.3 Extending the field of HAR An important advantage to generating virtual IMU data is that you can place the virtual data in wide variety of places on the human body. While some of the standard datasets we used in this work have subjects wearing multiple IMUs, there are limits to how many devices one can wear and still perform activities naturally. IMUTube removes that limitation. Now, for any given activity, we can experimentally determine where to place one IMU (or multiple IMUs) to best recognize that activity. For a set of activities, which place optimizes the recognition of all of the activities in that set. We have never had the ability to contemplate that kind of question. We also need not limit to IMUs placed directly on the body. Models of how clothing responds on a body might be used to generate virtual IMU data for objects that are loosely connected to the body [2, 60]. HAR can now inform clothing manufacturers of where in the material for a shirt, for example, you would want to integrate IMU data collection to predict the activities of the person wearing the shirt, or any other piece of clothing for that matter [36, 49].

6.3.4 Real IMU as 'seeds' to our pipeline While IMUTube is about generating lots of virtual IMU data, our results show the value for the more traditional curated datasets from real IMU data. The real IMU data provides a seed that the virtual data grows into more sophisticated HAR models. Now the efforts in real IMU data collection can be focused on producing very high quality labelled data from a wide enough variety of subjects performing key activities. It may even be the case that this real IMU seed data is the treasured commodity that companies can use to provide the best seeds for IMUTube-generated virtual IMU data and the models grown from them.

7 Related work

The proposed method details a pipeline towards opportunistically extracting virtual sensor data from a potentially very large body of publicly available videos. This is in contrast to current wearable sensor data collection protocols, which involve user studies and human participants, as well as other approaches that generate sensor data from motion capture (mocap) settings. In what follows, we first discuss approaches to data collection for sensor-based human activity recognition as well as mocap based techniques. These approaches represent the state-of-the-art in the field that are based on dedicated data recording protocols. Subsequently, we detail prior work on training classifiers with limited labelled data, thereby focusing on data augmentation techniques and transfer learning.

7.1 Sensor Data Collection in HAR

Sensor data collection for human activity recognition is performed by conducting user studies [13, 64, 85]. Typically, the participants in a study are asked to perform activities in laboratory settings while wearing a sensing platform. The advantage of data recording in a lab setting is that in addition to sensor data typically video data is recorded that is subsequently used for manual data annotation. For this purpose, the sensor and video data streams need to be synchronized [58], and human annotators need to be trained for consistency in annotation. The laboratory is designed to resemble a real-world environment, and user activities are either scripted or naturalistic. These include various gesture and locomotion level activities. However, designing a lab study to capture realistic natural behaviors is difficult. The protocol of such studies makes it challenging to collect large scale datasets. Furthermore, the annotation of activities is costly and error-prone and therefore prohibitive towards creating large datasets as they are required for deriving complex machine learning models.

Recently, Ecological Momentary Assessment (EMA) based approaches have been employed record and especially annotate real-world activity data [31, 41, 78]. The sensing apparatus (containing sensors such as accelerometers or full-fledged IMUs) is worn on-body, and users self-report the activity labels when they are asked to do so through direct notification. Although these methods may lose sample-precise annotation of the activities, they encourage the collection of larger scale datasets. While limited to gesture-based activities, Laput and Harrison [41] have shown that larger numbers (83) of fine-grained hand activities can be reliably recorded and annotated. Both in-lab and EMA based collection protocols directly involve human participants to collect movement data using body-worn sensors.

Other approaches have explored alternative data collection methods that do not directly involve human participants. Kang *et al.*render a 3D human model on computer graphics software and simulate human activities [35]. The sensor data is extracted from the simulated human motion, and subsequently used to train the recognition models. However, it is very difficult to realistically simulate and design complex human activities. Therefore, such methods typically only explore simple gestures and locomotion activities. Alternatively, [77, 81] extract sensory data from public, large-scale motion capture (mocap) datasets [1, 47, 52], which contain a variety of motions and poses for human activity recognition. Although these datasets cover hundreds of subjects and thousands of poses and motions, they rarely include everyday activities. The majority of such mocap datasets include dancing, quick locomotion transitions, and martial arts, which are less relevant to recognizing daily human activities.

In this work, we instead leverage the availability of large scale video datasets that cover real-world activities to extract sensory data. These videos are recorded in-the-wild and contain a wide range of activities, including everyday activities, which makes them very attractive for deriving realistic and robust human activity recognition systems.

7.2 Tackling the Sparse Data Problem

Many publicly available datasets for human activity recognition contain imbalanced classes. For example, approximately 75% of the Opportunity dataset (which has 18 classes in total) [13] consists of the null class [24], making it challenging to design classifiers. The activities being studied also impact the class imbalance to some extent. In the PAMAP2 dataset, the skipping rope class constitutes approximately 2.5% of data, relative to other activities which constitute around 9% on average [24]. This follows reason as it is harder for subjects to perform rope skipping for longer durations of time, in contrast to walking or lying down. This resulting class imbalance poses a challenge for the design and training of classifiers, which may find it easier to simply predict the majority class. Furthermore, the relatively small size of labelled datasets results in models quickly overfitting and does not allow application of complex model architectures. It is also difficult to apply potentially alleviating techniques such as transfer learning, which rely on large datasets for knowledge transfer.

As a way to overcome the problem of small, class-imbalanced datasets, data augmentation techniques have been applied previously to prevent overfitting, improve generalizability and increase variability in the datasets. They involve techniques that systematically transform the data during the training process in order to make classifiers more robust to noise and other variations [48]. They artificially inflate the training data by utilizing methods, which perform data warping, or oversampling [72]. Data warping includes geometric transformations such as rotations, and cropping, as well as adversarial training. Meanwhile, oversampling involves feature space augmentations and generative adversarial networks (GANs) [72]. For time series classification, the data warping techniques include window slicing, window warping, rotations, permutations and dynamic time warping [18, 42]. Several of these transformations can be combined to further improve the performance over a single method. Um *et al.* demonstrate that combining three basic methods (permutation, rotation and time warping) yields better performance than using a single method [80]. In [63], construction equipment activity recognition is also improved by combining simple transformations.

Oversampling based methods include synthetic minority oversampling technique (SMOTE) [20] which addresses the class imbalance problem by oversampling, and GANs, which are, for example, used for augmenting biosignals [27]. In [62], a data augmentation technique for time series data with irregular sampling is proposed utilizing conditional GANs. It is shown to outperform data warping techniques such as window slicing and time warping. Augmentation for wearable sensor data has been explored for monitoring ParkinsonâĂŹs disease in [80]. In this paper, seven transformations, including jittering, scaling, rotation and warping are detailed and their effect relative to no augmentation is studied. Further, the authors observed that combining multiple transformations results in higher performance. In [75], augmentation is performed on IMU spectrogram features to improve the activity recognition performance.

Another approach to deal with small labelled datasets includes transfer learning. Here, a base classifier (typically a neural network) is first trained on a base dataset and task. Subsequently, the learned features are re-purposed, or *transferred*, to a second target network to be trained on the target dataset and task. In particular, if the target dataset is significantly smaller compared to the base dataset, transfer learning enables training a large target network without overfitting [82], and typically results in improved performance. In [68], the authors propose a self-supervision pretext task and demonstrate its effectiveness for unsupervised transfer learning on other datasets with little labelled data. A more extreme example of having very small labelled datasets includes one-shot and few-shot learning, which contain very few labelled samples per class [19].

While the data augmentation techniques do improve the classification performance, they, ultimately, produce perturbed training samples. Therefore, they are unable to provide for the variety in human movements that is obtained by collecting data from a large number of subjects. On the other hand, the GAN based techniques perform augmentation by sampling from the dataset distributon. However, they require substantial amounts of data to train, and suffer from training instability and non-convergence. Furthermore, there is limited prior work studying data augmentation by GANs for wearable sensor data and their actual suitability for sensor-based human activity recognition remains to be shown. This makes it challenging to readily apply these generative networks to create more data.

We tackle the problem of having small labelled datasets with a different approach – by generating large quantities of virtual IMU data from videos. As we can leverage a large body of videos, containing many individuals, we generate datasets containing more diverse movements and potentially much larger datasets of realistic data, which is in stark contrast to existing methods that try to combat the sparse data problem.

8 Conclusion

In this paper we proposed the idea of generating virtual IMU data based on automated extraction from video as a means to collect large-scale labelled dataset to support research in human activity recognition (HAR). We designed and validated an initial proof of concept prototype, IMUTube, that integrates a collection of techniques from computer vision, signal processing, and machine learning. Our initial findings show great promise for this technique to extend the capabilities for HAR, at a minimum for simple activities whose main IMU characteristics are confined to expression in 2D.

The greater promise of this work requires a collective approach by computer vision, signal processing, and activity recognition communities (who have already been greatly united through the advances of deep learning) to advance our proposed agenda. Computer vision researchers can clearly build upon the IMUTube pipeline to address a variety of current limitations, further automating the pipeline and reducing the need for human curation of online videos. Signal processing advances can further manipulate the virtually-generated data to better condition the virtual data and represent the features and distributions of real IMU data. Activity recognition researchers can apply known modern learning techniques to this new class of labelled data for HAR and develop more effective ways to model, both with and without a mixture of real IMU data. Within a few years, we expect this collective effort to result in HAR as yet another success story for large-data-inspired learning techniques.

References

[1] 2008?. Carnegie Mellon Motion Capture Database. Retrieved April 25, 2020 from http://mocap.cs.cmu.edu/

[2] T. Alldieck, M. Magnor, B. Bhatnagar, C. Theobalt, and G. Pons-Moll. 2019. Learning to reconstruct people in clothing from a single RGB camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1175–1186.

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] P. Asare, R. Dickerson, X. Wu, J. Lach, and J. Stankovic. 2013. BodySim: A Multi-Domain Modeling and Simulation Framework for Body Sensor Networks Research and Design. ICST.

[5] M. Bächlin, M. Plotnik, and G. Tröster. 2010. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Trans. Inf. Technol. Biomed.* 14, 2 (2010), 436–446.

[6] P.J. Besl and N. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (feb 1992), 239–256.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. 3464–3468.

[8] O. Bogdan, V. Eckstein, F. Rameau, and J. Bazin. 2018. DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production, CVMP 2018, London, United Kingdom, December 13-14, 2018*. ACM, 6:1–6:10.

[9] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*. 961–970.

[10] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7291–7299.

[11] B. Caprile and V. Torre. 1990. Using vanishing points for camera calibration. *International Journal of Computer Vision* 4, 2 (mar 1990), 127–139.

[12] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987* (2019).

[13] R. Chavarriaga, H. Sagha, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.* 34, 15 (2013), 2033–2042.

[14] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005* (2013).

[15] W. Conover and R. Iman. 1981. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician* 35, 3 (1981), 124–129.

[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[17] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[18] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455* (2018).

[19] Siwei Feng and Marco F Duarte. 2019. Few-shot learning-based human activity recognition. *Expert Systems with Applications* 138 (2019), 112782.

[20] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research* 61 (2018), 863–905.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[22] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. 2019. Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

[23] C. Gu, C. Sun, D. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6047–6056.

[24] Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–28.

[25] N. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 international symposium on wearable computers*. 65–68.

[26] N. Y. Hammerla, S. Halloran, and T. Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables.. In *IJCAI*. AAAI Press, 1533–1540.

[27] Shota Haradal, Hideaki Hayashi, and Seiichi Uchida. 2018. Biosignal data augmentation based on generative adversarial networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 368–371.

[28] H. Haresamudram, D. Anderson, and T. Plötz. 2019. On the role of features in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*. 78–88.

[29] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[30] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29, 6 (2012), 82–97.

[31] K. Hovsepian, M. Al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar. 2015. cStress: towards a gold standard for continuous stress assessment in the mobile environment. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*. 493–504.

[32] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[33] I. Joel, A.and Stergios. 2011. A Direct Least-Squares (DLS) method for PnP. In *2011 International Conference on Computer Vision*. IEEE.

[34] A. Kanazawa, M. Black, D. Jacobs, and J. Malik. 2018. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7122–7131.

[35] C. Kang, H. Jung, and Y. Lee. 2019. Towards Machine Learning with Zero Real-World Data. In *The 5th ACM Workshop on Wearable Systems and Applications*. 41–46.

[36] S. Kang, H. Choi, H. Park, B. Choi, H. Im, D. Shin, Y. Jung, J. Lee, H. Park, S. Park, and J. Roh. 2017. The development of an IMU integrated clothes for postural monitoring using conductive yarn and interconnecting technology. *Sensors* 17, 11 (2017), 2560.

[37] P. Karlsson, B. Lo, and G. Z. Yang. 2014. Inertial sensing simulations using modified motion capture data. In *Proceedings of the 11th International Conference on Wearable and Implantable Body Sensor Networks (BSN 2014), ETH Zurich, Switzerland*. 16–19.

[38] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[39] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. 2011. HMDB: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*. IEEE, 2556–2563.

[40] Nicholas D. Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T. Campbell, and Feng Zhao. 2011. Enabling Large-Scale Human Activity Inference on Smartphones Using Community Similarity Networks (Csn). In *Proceedings of the 13th International Conference on Ubiquitous Computing* (Beijing, China) *(UbiComp âĂŽ11)*. Association for Computing Machinery, New York, NY, USA, 355âĂŞ364. https://doi.org/10.1145/2030112.2030160

[41] G. Laput and C. Harrison. 2019. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[42] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks.

[43] W. Li, Z. Zhang, and Z. Liu. 2010. Action recognition based on a bag of 3D points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 9–14.

[44] D. Liaqat, M. Abdalla, Pegah Abed-Esfahani, Moshe Gabel, Tatiana Son, Robert Wu, Andrea Gershon, Frank Rudzicz, and Eyal De Lara. 2019. WearBreathing: Real World Respiratory Rate Monitoring Using Smartwatches. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–22.

[45] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Duan, and A. Kot. 2019. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). https://doi.org/10.1109/TPAMI.2019.2916873

[46] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 1–16.

[47] N. Mahmood, N. Ghorbani, N. Troje, G. Pons-Moll, and M. Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE International Conference on Computer Vision*. 5442–5451.

[48] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, Leonid Joffe, Nicholas D Lane, Fahim Kawsar, and Pietro Lió. 2018. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 200–211.

[49] A. Muhammad Sayem, S. Hon Teay, H. Shahariar, P. Fink, and A. Albarbar. 2020. Review on Smart Electro-Clothing Systems (SeCSs). *Sensors* 20, 3 (2020), 587.

[50] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[51] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and checkerboard artifacts. *Distill* 1, 10 (2016), e3.

[52] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. 2013. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 53–60.

[53] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[54] F. J. Ordóñez and D. Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[55] J. Park, Q. Zhou, and V. Koltun. 2017. Colored Point Cloud Registration Revisited. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 143–152.

[56] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7753–7762.

[57] T. Phạm and Y. Suh. 2018. Spline Function Simulation Data Generation for Walking Motion Using Foot-Mounted Inertial Sensors. In *Sensors*. Electronics, 199–210.

[58] T. Plotz, C. Chen, N. Hammerla, and G. Abowd. 2012. Automatic synchronization of wearable sensors and video-cameras for ground truth annotation–a practical approach. In *2012 16th international symposium on wearable computers*. IEEE, 100–103.

[59] F. Pomerleau, F. Colas, and R. Siegwart. 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends Robot* 4, 1 (May 2015), 1âĂŞ104.

[60] G. Pons-Moll, S. Pujades, S. Hu, and M. Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–15.

[61] G. Pons-Moll, J. Romero, N. Mahmood, and M> Black. 2015. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–14.

[62] Giorgia Ramponi, Pavlos Protopapas, Marco Brambilla, and Ryan Janssen. 2018. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295* (2018).

[63] Khandakar M Rashid and Joseph Louis. 2019. Times-series data augmentation and deep learning for construction equipment activity recognition. *Advanced Engineering Informatics* 42 (2019), 100944.

[64] A. Reiss and D. Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*. IEEE, 108–109.

[65] A. Reiss and D. Stricker. 2013. Personalized mobile physical activity recognition. In *Proceedings of the 2013 international symposium on wearable computers*. 25–28.

[66] M. Rosca, B. Lakshminarayanan, and S. Mohamed. 2018. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847* (2018).

[67] S. Rusinkiewicz and M. Levoy. [n.d.]. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE Comput. Soc.

[68] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.

[69] P. M. Scholl, M. Wille, and K. Van Laerhoven. 2015. Wearables in the wet lab: a laboratory system for capturing and guiding experiments. In *Ubicomp*. ACM, 589–599.

[70] S. Shah and J.K. Aggarwal. 1996. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition* 29, 11 (nov 1996), 1775–1788.

[71] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao. 2019. Human-Aware Motion Deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*. 5572–5581.

[72] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 60.

[73] G. Sigurdsson, G. Varol, X. Wang, I. Laptev, A. Farhadi, and A. Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. *ArXiv e-prints* (2016). arXiv:1604.01753 http://arxiv.org/abs/1604.01753

[74] K. Soomro, A. Zamir, and M. Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[75] Odongo Steven Eyobu and Dong Seog Han. 2018. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors* 18, 9 (2018), 2892.

[76] T. Sztyler and H. Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–9.

[77] S. Takeda, T. Okita, P. Lago, and S. Inoue. 2018. A multi-sensor setting activity recognition simulation tool. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 1444–1448.

[78] E. Thomaz, I. Essa, and G. Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1029–1040.

[79] Matt Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *2017 British Machine Vision Conference (BMVC)*.

[80] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinsonâĂŹs disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 216–220.

[81] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li. 2020. A Deep Learning Method for Complex Human Activity Recognition Using Virtual Wearable Sensors. *arXiv preprint arXiv:2003.01874* (2020).

[82] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.

[83] A. Young, M. Ling, and D. Arvind. 2011. IMUSim: A simulation environment for inertial sensing algorithm design and evaluation. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks, IPSN 2011, April 12-14, 2011, Chicago, IL, USA*. IEEE, 199–210.

[84] J. Yu and R. Ramamoorthi. 2019. Robust Video Stabilization by Optimization in CNN Weight Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3800–3808.

[85] M. Zhang and A. A. Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Ubicomp*.

[86] Q. Zhang and R. Pless. [n.d.]. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. IEEE.

[87] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. 2011. Cross-people mobile-phone based activity recognition. In *Twenty-second international joint conference on artificial intelligence*.

[88] T. Zhou, M. Brown, Noah S., and D. Lowe. 2017. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1851–1858.

[89] H. Zhuang. 1995. A self-calibration approach to extrinsic parameter estimation of stereo cameras. *Robotics and Autonomous Systems* 15, 3 (aug 1995), 189–197.