CROMOSim: A Deep Learning-based Cross-modality Inertial Measurement Simulator

Yujiao Hao, Xijian Lou, Boyu Wang, and Rong Zheng, *Senior Member, IEEE*

Abstract—With the prevalence of wearable devices, inertial measurement unit (IMU) data has been utilized in monitoring and assessing human mobility such as human activity recognition (HAR) and human pose estimation (HPE). Training deep neural network (DNN) models for these tasks require a large amount of labelled data, which are hard to acquire in uncontrolled environments. To mitigate the data scarcity problem, we design CROMOSim, a cross-modality sensor simulator that simulates high fidelity virtual IMU sensor data from motion capture systems or monocular RGB cameras. It utilizes a skinned multi-person linear model (SMPL) for 3D body pose and shape representations to enable simulation from arbitrary on-body positions. Then a DNN model is trained to learn the functional mapping from imperfect trajectory estimations in a 3D SMPL body tri-mesh due to measurement noise, calibration errors, occlusion and other modelling artifacts, to IMU data. We evaluate the fidelity of CROMOSim simulated data and its utility in data augmentation on various HAR and HPE datasets. Extensive empirical results show that the proposed model achieves a 6.7% improvement over baseline methods in a HAR task.

Index Terms—deep learning, IMU, simulation, human activity recognition, human pose estimation

1 Introduction

Nowadays, inertial measurement units (IMUs) have become ubiquitously available in wearable and mobile devices. An important category of IMU-enabled applications is monitoring and assessing human mobility, which aims to continuously track people's daily activities, analyze motion patterns and extract digital mobility bio-markers such as gait parameters in the wild. Increasingly, data-driven deep learning models have been developed for human activity recognition (HAR) [1], [2] and human pose estimation (HPE) [3]. Despite their impressive performances, these models generally require a large amount of sensory data for model training. Unfortunately, it is challenging to collect high-quality IMU data in the wild. Moreover, data collected from controlled settings where subjects are asked to perform certain activities often have very different characteristics from those in freestyle motions [4]. On the other hand, annotating IMU data post hoc is challenging as raw IMU signals are hard to interpret even by domain experts.

The scarcity of IMU data for human mobility assessment is evident when compared with the richness of other data sources. PAMAP2 [5], a benchmark dataset for HAR, consists of 8 subjects with only 59.67 minutes of samples per person. In contrast, AMASS [6], a motion capture (MoCap) dataset, includes 2420.86 minutes of data and is still growing; not to mention YouTube videos, which offer a practically infinite amount of action data. Therefore, to mitigate the "small data" problem, one possible solution is to convert data from other modalities to IMU, a process called *cross-modality simulation*.

Though several previous works explored the feasibility of simulating IMU sensor data from other data modalities (see Section 2), two fundamental challenges remain. First, sensors are attached to human skin rather than directly to bone joints during data collection. Skeleton models are inadequate in representing human poses and shapes. Second, even with state-of-the-art (SOTA) solutions in computer vision, the extracted 3D human motion trajectories from monocular video clips remain inaccurate. Analytically compute IMU readings on such imperfect input sequences will result in large errors. However, if a deep learning model is adopted to learn the mapping between noisy motion trajectories and measured sensor readings, it is unclear how well such models generalize to arbitrary unseen on-body positions.

To tackle the aforementioned challenges, we design and implement CROMOSim, a cross-modality IMU sensor simulator that simulates high fidelity virtual IMU sensor data from motion capture systems and monocular RGB cameras. It differs from existing work in two important aspects. First, it is based on the 3D skinned multi-person linear (SMPL) model [7], which serves as an intermediate representation of motion sequences and entitles our CROMOSim for an arbitrary on-body simulation. SMPL has been widely used in HPE tasks [8], [9], [10], [11], which is capable of modelling muscle and soft tissue artifacts. In contrast, the 2D or 3D skeleton representations adopted by other works are segment models without volumetric information. Second, we empirically demonstrated that the direct computation of IMU readings from motion trajectories extracted from videos is unreliable (in Section 4), even with filtering and interpolation techniques as the case of IMUSim [12]. We instead design and train a neural network to learn the relationship between measured IMU readings and the noisy

- *Yujiao Hao, Xijian Lou and Rong Zheng are with the Department of Computing and Software, McMaster University, Hamilton, Canada. E-mail: haoy21@mcmaster.ca; loux10@mcmaster.ca; rzheng@mcmaster.ca.*
- *Boyu Wang is with Department of Computer Science, Western University, London, Canada. E-mail: bwang@csd.uwo.ca.*

motion trajectories. Special cares have to be given to ensure the trajectories are represented in a consistent global coordinate frame even if the videos are captured by moving cameras. Compared to existing IMU simulators, experiments show that CROMOSim achieves higher fidelity and superior performance in various HAR tasks. HPE tasks are also evaluated to demonstrate the utility of simulated data in downstream applications.

In summary, we make the following contributions in this paper:

1) CROMOSim is the first work that utilizes SMPL full-body tri-mesh as an intermediate representation for IMU data simulation.
2) CROMOSim offers a generic pipeline for generating IMU readings at *arbitrary* on-body locations from either MoCap or monocular RGB data. It is readily extensible to other input modalities and configurations.
3) CROMOSim mitigates imperfection in intermediate body pose and shape estimations through a supervised learning approach.
4) Compared to SOTA IMU simulators, CROMOSim achieves higher fidelity and superior performance in HAR tasks.
5) We are the first to empirically show the utility of simulated IMU data in HPE tasks under a deep learning context.

The rest of the paper is organized as follows. Section 2 describes related work. In Section 3, we introduce the CROMOSim pipeline and details of each component. In Section 4 we present the implementation details and performance evaluation of CROMOSim standalone and in downstream tasks. Finally, we conclude the paper in Section 5 with discussion and future work.

2 Related Work

The proposed cross-modality simulation framework is a type of data augmentation technique, which is broadly used in machine learning to compensate for data scarcity, to improve data diversity, and boost the generalization of a trained model. In the context of augmenting IMU data, we categorize existing methods into three groups: transforming real IMU recordings, generative models for IMU data, and cross-modality simulators.

IMU transformations: Simple operations such as flipping, rotation, scaling and changes in brightness can be applied to augment image data. Similar ideas are applicable to IMU data as well. In [13], [14], random relative rotations between a sensor and human body were added within a predefined range, to make the trained model robust to subject divergence. In [15], the authors proposed a systematic way to augment the IMU data via rotation, permutation, time-warping, scaling, magnitude-warping and jittering. Eyobu *et al.* went even further in [16] to transform handcrafted features rather than the raw recordings of wearable sensors. Though easy to implement, IMU transformation methods require the availability of sufficient real sensor data as their source.

Generative models for IMU data: Generative adversarial networks (GAN) use two neural networks, pitting one against the other in order to generate new, synthetic instances of data that are indistinguishable from real ones [17]. Researchers designed GANs to generate IMU data in [18], [19], [20]. In [19], a conceptual solution was proposed. SensoryGANs [18] adopt adversarial learning in generating diverse yet realistic IMU sensor readings for locomotion. However, this method is highly complex: a different neural network architecture is devised for each activity. Moreover, due to the large variances in the generated data, simulated data can not be translated back to meaningful human motion trajectories, thus making it only suitable for relatively simple HAR tasks with easily separable patterns. For example, both SensoryGANs and ActivityGANs [20] simulate only *stay still, walk, jog* activities in their evaluation. Another limitation of GAN-based methods is that they tend to generate data that is similar to real data in the training set and are not reliable to produce data for new subjects or new activities.

Cross-modality IMU simulation: Given motion trajectories in a global frame, acceleration can be calculated by taking the second derivatives of positions over time. Researchers may take advantage of this simple computation strategy to simulate accelerometer data from MoCap sequences. The resulting data has been used in recent works to pre-train human pose estimation (HPE) [21] and HAR models [22], [23]. One drawback of this method is that none of these researches targets to simulate realistic IMU sensor readings, and gyroscope data is omitted. For a more systematic IMU simulation, IMUSim [12] is among the first open-source tool to simulate IMU data from either MoCap data in the Biovision Hierarchy (BVH) format or a user-provided 3D position and orientation sequence. Though employs data smooth and filtering techniques to tackle outliers, this method is built upon analytically calculation with low data fidelity (see Section 4.2).

After that, simulating IMU readings from monocular RGB videos for data augmentation has attracted some attention in recent years. ZeroNet [24] extracted finger motion data from videos, then transformed them into acceleration and orientation information measured by IMU sensors. The authors of [25] and its follow-up work [26], [27] simulated acceleration norms and/or angular velocity norms from human 2D poses for a HAR purpose. In their latest work [27], Rey et al. skipped the video processing steps and directly mapped vision data to IMU readings with placement specific neural networks. These works avoid the video-based global motion tracking by limiting human subjects' movement to a fixed camera scene (in-place motion), and thus cannot be applied to handle in-wild video data with moving cameras. Closest to our work are IMUTube [28] and its extension in [29], which aim at simulating full-body IMU data from moving camera videos captured in the wild. But limited by the skeleton body representation adopted, neither work can simulate realistic sensor readings from arbitrary on-body locations. Moreover, in IMUTube, the estimation of view depth and camera ego-motion is in two independent steps though the two are intrinsically coupled [30], [31]. A wrongly predicted camera pose can lead to inaccurate view depth estimation and vice versa [32]. In addition, the lifting of 2D postures to 3D module in IMUTube pipeline is more compute-intensive and error-prone, as it is a simple combination of existing technologies.

3 METHOD

Before introducing the proposed method, the notations used in this chapter are defined as follows. There are four different coordinate frames involved in this work: F^G for the global tracking frame, which is a fixed coordinate system for representing objects in the world; F^B for a bone coordinate frame, which originates at the bone's distal joint with x-axis along the bone pointing to its proximal joint and y-axis in the medal-lateral direction; F^S is the sensor frame that is fixed on the sensor and is determined by its manufacturer; F^C for the camera frame that takes the center of the camera's image plane as its origin and the optic axis as the Z-axis (Fig. 1). Rotation matrix R^S_B denotes the rotation from bone frame to sensor frame. For simplicity, amongst camera intrinsic parameters, we assume the optical centers of the camera in pixels on the x and y axis $c_x = c_y = 0$, and only estimate the focus length in the x and y axis f_x and f_y. Camera extrinsic parameters include rotation matrix R and translation vector t, respectively. R and t are fixed for fixed cameras and need to be updated for moving cameras. During movements, both F^B and F^S changes relative to F^G and are placement or device specific. Therefore, it is necessary to transform representations of motions into a unified global coordinate first.

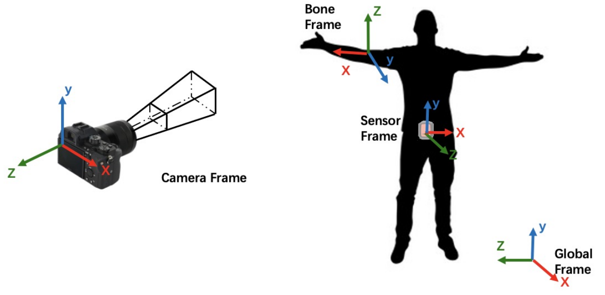


Fig. 1: An example of different coordinate frames involved in this work.

3.1 Overview

CROMOSim is designed with several requirements in mind: i) allowing arbitrary user-specified placement and orientation of target sensors, ii) extensibility to different input data modalities and configurations, iii) flexibility to incorporate SOTA models to extract motion trajectories, and iv) high fidelity. To meet these requirements, the CROMOSim pipeline consists of three function modules as shown in Fig. 2 : an input data processing module that extracts global human motion sequences from source data, a human body model that fully represent the extracted sequences and can be sampled from any on-body location, and a simulator module that transforms noisy motion sequences into high-fidelity 3-axis accelerometer and gyroscope readings. Though the pipeline is extensible to other possible input data modalities such as millimetre wave radar and depth camera, we will focus on MoCap and monocular camera video here. Each component will be discussed in detail in the remaining section.

3.2 SMPL Model

An SMPL model represents 3D human body poses and shapes with a fine-grained full-body tri-mesh. Unlike skeleton or cylinder models that only capture joint poses, this parametric 3D representation provides a widely applicable and differentiable way to visualize a realistic 3D human body. There are three reasons to choose SMPL over other body models in CROMOSim. First, instead of measuring the movements of bones, IMU readings reflect the soft tissue dynamics at the location to where a sensor is attached. Second, SMPL provides a pose and shape-dependent full-body tri-mesh that can be sampled at any on-body location. Third, since it is widely used in HPE research, many off-the-shelf models are available to extract SMPL representations from different data sources.

To see the difference between movements of joints in a skeleton model and SMPL skin mesh, we compare accelerations computed by taking second-order derivatives of the corresponding motion trajectories and ground-truth accelerometer readings over time. In Fig. 3, red curves denote the calculated 3-axis accelerations while the black ones are accelerometer ground truth. Figures in the left column compare the accelerations at the pelvis joint in a skeleton model while figures in the right column compare those at SMPL lower back skin mesh vertices. Clearly, the use of the SMPL skin mesh provides better agreements with the ground truth (e.g., in the interval [100,300]). Simulated data from the pelvic joint, on the other hand, fails to capture high-frequency acceleration components, which are most likely due to muscle and soft issue movements. SMPL enables users to sample from any on-body position on the skin surface while the skeleton model represents the motion of bones only. In most cases, IMUs are attached to body surfaces rather than directly to bones or anatomical landmarks. Thus, SMPL is a good candidate for an intermediate data representation of the CROMOSim pipeline.

3.3 Input Data Processing

3.3.1 From MoCap Data to SMPL Models

MoCap data consists of raw marker sequences collected by an optical motion capture system of high precision (usually with a position error < 1 mm). With commercial Mocap systems like OptiTrack [33] and Vicon [34], both body shape and pose data can be captured. Such data have been widely used as ground truth labels in markerless human pose estimation with cameras or wearable sensors [3]. MoSH++ [6] allows the fit of an SMPL model to MoCap data from a set of sparse markers. Prior to motion capture, a global tracking coordinate system needs to be established during the calibration phase. As a result, the collected motion trajectories are expressed in the defined global frame. Under the assumption that the global frame is aligned with the

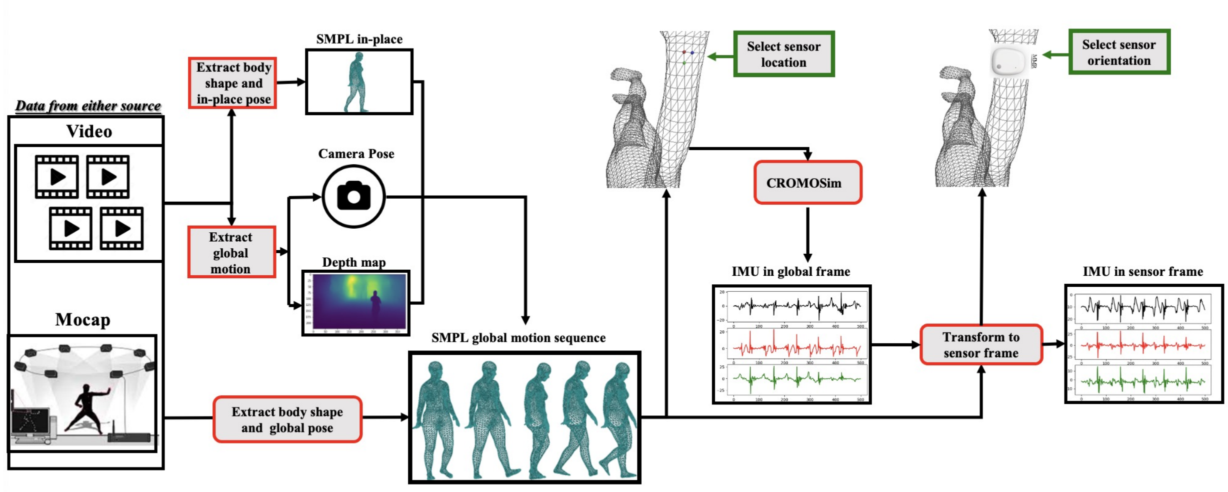


Fig. 2: The proposed CROMOSim pipeline. It takes either MoCap or monocular camera video data as input and converts them into SMPL represented global motion and body shape. The simulator then takes the SMPL model, specified sensor placement and orientation as input; predicts simulated IMU readings and transforms them back to the sensor coordinates frame.

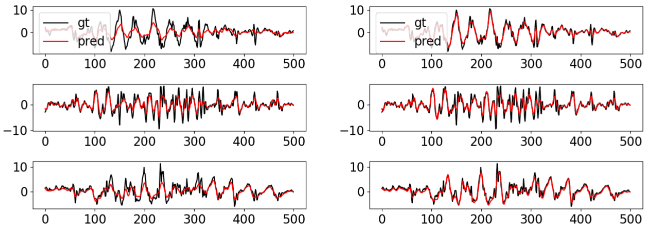


Fig. 3: Comparison between analytically computed 3-axis accelerations from a skeleton representation and an SMPL model. Left: taking the motion sequence of pelvis joint positions as input, right: taking the motion sequence of SMPL lower back skin mesh positions as input.

inertial frame[1] , the SMPL mesh model can be used directly in subsequent processing.

3.3.2 From Video Clips to SMPL Models

Extracting 3D human poses and shapes from monocular RGB videos is not trivial, especially when they are captured from moving cameras with unknown parameters, which is common in a locomotion-related video recorded in the wild. We propose to decompose such a problem into two subproblems: a reconstruction of human global displacement and rotation; and an estimation of 3D in-place human motion and body shape.

Estimating root joint global trajectory: A precise calculation of global displacement for the human subject is essential for a high-fidelity simulation of IMU data from RGB videos. This requirement can be achieved by reconstructing the 3D motion trajectory of a fixed body position (a.k.a, the root joint), which can be inferred from the per frame depth map of the human subject and known camera parameters [35].

In CROMOSim, we adopt robust consistent video depth estimation (Robust CVD) method [30], a SOTA model to estimate consistent dense depth maps and camera poses from a monocular video. Robust CVD jointly estimates both outputs by solving an optimization problem over the entire video sequence. It is advantageous as the two outputs are intrinsically coupled and thus lead to higher accuracy (compared to the pipeline adopted by IMUTube). In the implementation, we locate the 2D torso joint positions in video frames using OpenPose [36], and designate the pelvis as our root joint. With the detected 2D joint position and depth map per video frame, we can calculate the global 3D torso coordinates as follow. Denote the 3D coordinates of the root joint in the camera frame and the global frame at time k by $P^C(k) = [X^C(k), Y^C(k), Z^C(k)]$ and $P^G(k) = [X^G(k), Y^G(k), Z^G(k)]$ respectively. Let its corresponding 2D pixel coordinates in the camera image be $[x(k), y(k)]$. Given the camera intrinsic parameters f_x and f_y from robust CVD, we have

$$\begin{aligned} X^C(k) &= \frac{(x(k) - \frac{W}{2}) \times Z^C(k)}{f_x} \\ Y^C(k) &= \frac{(y(k) - \frac{H}{2}) \times Z^C(k)}{f_y} \\ Z^C(k) &= d(x(k), y(k)), \end{aligned} \tag{1}$$

where $d(x, y)$ is a depth retrieving function with a 2D pixel coordinates x, y, and W and H are the width and height of the pixel image. Next, using the camera extrinsic parameters R_k and t_k, we transform the root joint position from the camera frame F^C to global frame F^G at time k follows:

$$P^G(k) = R_k^T \times (P^C(k) - t_k) \tag{2}$$

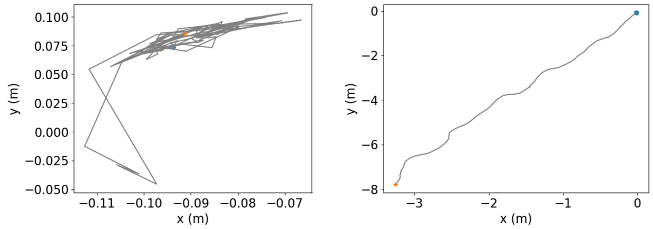
In addition, depth reconstructed by robust CVD is reasonably accurate up to scale. To resolve scale ambiguity, an object of known size (its real height h_r or real width

[1] Such an assumption is not restrictive as a random rotation can be applied in further data augmentation to obtain data if the global and inertial frames differ.

w_r) in the scene is needed, as real depth at time k can be calculated with $d_r(k) = (f_y \times h_r)/h_p(k)$, where $h_p(k)$ is the object height in pixels. The scale factor can be estimated with $s = d_r(k)/d(x(k), y(k))$, and it is a constant value per video clip processed by Robust CVD. Prior knowledge regarding heights of subjects in the video, or dimensions of fixtures (e.g., street lamps, road lanes) can be utilized. Subsequently, the predicted depth of the pelvis joint is re-scaled by the estimated scale factor to recover the real global root joint trajectory.

Since in some frames, the root joint is not visible or cannot be located accurately due to occlusion or poor lighting, we only extract root joint coordinates from the frames with high confident scores by OpenPose. Root joint coordinates in the remaining frames are then interpolated from the estimated ones, and a Kalman filter is applied to further smooth the resulting trajectory.

Body pose and shape estimation in camera frames: We adopt VIBE [9], a SOTA method to directly estimate realistic 3D human poses and shapes from monocular videos. In the implementation, we make two extensions to VIBE. First, VIBE assumes a fixed camera configuration and in-place human motion only, losing track of human subjects' global motion trajectory. Fig. 4 shows the difference between motion trajectories of a lower back SMPL mesh vertex near a subject's pelvis. The figures are extracted by VIBE only, and by our proposed pipeline, respectively, when the straight-line running subject was captured by a handheld camera. Clearly, the trajectory in the left figure fails to reflect the actual motion. As elaborated in the previous paragraph, robust CVD is adopted to complete the missing information. It helps to estimate the 3D global translation of the subject's root joint per video frame even when there is relative motion between the camera and the human subject. We acquire a full 3D human pose representation by adding the global translation to the translation parameters of the SMPL model from VIBE outputs. Second, VIBE estimates body shapes for every video frame and a frequent re-scaling of the human subjects can be observed when there are drastic motions or the camera is fast moving. This is unnecessary since people's body shapes are unlikely to change in a short period and such rescaling is prone to errors. Instead, we assume that the estimated body shape can be modeled as a ground truth shape plus zero-mean random noise. Thus, shape estimation errors can be mitigated by averaging the estimated body shapes for the same subject over multiple frames in a 10-second video sequence.



x

Fig. 4: Extracted motion trajectories of a subject's low back from a 4-seconds running outdoor video clip captured with a handheld camera. The subject in the video runs along a straight line. Left: results from VIBE only, right: results from the proposed pipeline. Each plot was generated by projecting 3D trajectory on the ground plane.

Finally, by combining the aforementioned steps, we can extract 3D body poses in a global frame and shape parameters from monocular RGB video, which can serve as input to generate SMPL body meshes.

3.4 From SMPL Models to IMU Data

Given the 3D human pose and shape represented by SMPL tri-mesh over time, accelerations and angular velocities in a global frame can be computed analytically. In particular, accelerations can be calculated by taking second derivatives of positions over time; angular velocities can be determined from the changes in the normal vector of a plane associated with three non-collinear mesh points (e.g., the vertices of a mesh triangle). However, SMPL tri-meshes generated by the models in Section 3.3.2 tend to be noisy, erroneous and incomplete. Furthermore, accelerations and angular velocities measured by IMUs are subject to hardware imperfection such as noises, biases, and non-orthogonal axes, which are not easily replicated by analytical calculation.

To address the aforementioned issues, we design two neural network models, an accelerometer and a gyroscope network, to learn the mapping between motion trajectories of SMPL tri-mesh points and actual acceleration or angular velocity measured by IMUs in a global frame, respectively. The neural networks are capable of generating data from any arbitrary unseen region over the human body by training with real data from some selected on-body positions of various motion ranges (such as the head, chest, one side of the wrist, and ankle). Both models take the same design, with three convolutional and two bidirectional long-short term memory (LSTM) layers as the feature extractor, and a following linear layer for regression output. The model is fed a user-specified skin area, with three mesh triangles chosen near the area's center as input. In each triangle, the vertices are traversed counter-clockwise to ensure the norm direction always points outside of the human body.

The collected IMU data are usually in the local sensor frame while the predictions of CROMOSim are in the global frame. Therefore, a coordinates transformation step is required. A user needs to select the skin region a virtual sensor affixes to and define its alignment represented as a rotation matrix (R_S^B). With the rotation matrix from the bone frame to the sensor frame $(R_S^B)^{-1}$, we can transform IMU data into the sensor frame from the accelerations $\mathbf{a}_G$ and angular velocities ω_G in the global frame as follows:

$$\mathbf{a}_S = (R_S^B)^{-1} \times (R_B^G)^{-1} \times (\mathbf{a}_G + g), \tag{3}$$

and

$$\omega_S = (R_S^B)^{-1} \times (R_B^G)^{-1} \times \omega_G, \tag{4}$$

where g is the gravity acceleration and R_B^G is obtained from the SMPL model for the corresponding skin region.

Due to noisy data sources and modelling errors, domain gaps exist between simulated and real data. Such gaps are more pronounced in the simulated data from videos. To mitigate these gaps, we adopt the same distribution mapping technique [37] as IMUTube. Let $G(X \leq x_r)$ and $F(X \leq x_s)$ be the cumulative density functions (CDF) for real IMU x_r and simulated data x_s, respectively. Under the assumption that $G(\cdot)$ is invertible, it can be proven that $x'_s = G^{-1}(F(X \leq x_s)$ follows the same distribution as x_r.

To apply distribution mapping, we need to estimate the CDF of simulated and real data along each axis, then apply the mapping separately. Empirical results from IMUTube show that a small number of real data ($\sim$ 1000 samples per class or equivalently 33-second long at a sampling rate of 30 Hz) are sufficient to give a good estimation of $G(\cdot)$.

4 Evaluation

In this section, we will evaluate CROMOSim in two sets of experiments. Firstly, we evaluate the fidelity of simulated sensor data both qualitatively and quantitatively. Then, we evaluate the utility of CROMOSim in data augmentation for downstream HAR and HPE tasks.

4.1 Experimental Setup

4.1.1 Datasets

To train the simulator network and evaluate the fidelity of simulated data, we use the TotalCapture dataset, a benchmark for 3D HPE from marker-less multi-camera capture [38] which has all three data modalities (MoCap, IMU and video). For HAR evaluation, Realworld [39], the Physical Activity Monitoring version 2 (PAMAP2) [5] and Opportunity [40] datasets are used in task model training and testing. For knee angle estimation tasks, we also take Totalcapture in our experiments. A detailed description of each dataset is listed below:

1) **TotalCapture [38]:** It is the first dataset to have fully synchronized multi-view video collected from eight RGB cameras at a frame rate of 60Hz, 12 IMU sensors (affixed to a subject's head, right and left upper arms, right and left wrists, right and left upper legs, right and left lower legs, right and left feet and pelvis) sampled at 60Hz and Vicon labels for a large number of frames (~1.9M). It contains 5 subjects performing *acting, walking, rolling arms, and freestyle motions* indoor.
2) **Realworld [39]:** It has 8 activities including *climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking* performed by 15 subjects. Each subject wore mobile devices on 7 body positions (chest, forearm, head, shin, thigh, upper arm and waist). Videos were recorded by a moving handheld camera followed the subjects. Each activity lasted 10 minutes, except for jumping, which was around 2-minute long. Data was collected naturally. In some indoor trials, the light conditions were poor. In some outdoor trials, the videos contain passers-by not part of the subject pool.
3) **PAMAP2 [5]:** The Physical Activity Monitoring version 2 (PAMAP2) consists of data collected from IMU sensors (accelerometer and gyroscope) on subject's chest, dominant ankle and wrist during 8 activities, i.e., *lying, sitting, walking, running, standing, rope jumping, ascending stairs and descending stairs*. Eight subjects performed these activities freely without time constraints and had the option to skip some activities. There exist missing classes in some subjects' data and the data samples are unbalanced across the classes. During data collection, IMU sensors are instrumented on different subjects at a sampling rate of 100Hz.
4) **Opportunity [40]:** The Opportunity dataset contains IMU measurements from 4 subjects during 5 mobility-related activities. The activities are *sitting, standing, lying, walking and null*, where 'null' include any activity outside the first four. Data was collected from 7 body-mounted sensors (left and right forearms, left and right arms, back, left and right feet) at a sampling rate of 30Hz.

4.1.2 Data Preprocessing

In the fidelity evaluation, we divide data from TotalCapture with all modalities into 2-seconds sliding windows with 80% overlapping for model training and without overlapping for prediction. For HAR, to make the results directly comparable to baseline approaches, we follow the same procedure described in IMUTube, where simulated and real IMU data are low-pass filtered, normalized and divided into sliding windows with 1-second length and 50% overlapping. In the case of HPE, the real and simulated IMU data are standardized, and then divided into 1-second windows without overlapping.

4.1.3 Evaluation Metrics

To evaluate the fidelity of CROMOSim, we compute the root mean square error (RMSE) between simulated IMU data and ground truth. In HAR tasks, as the classes in datasets are imbalanced, we use F1 score to evaluate the random single-subject-out experiments. In multi-class classification, the F1 score is computed as the weighted average of the F1 score of each class. In 3D HPE tasks, we measure the RMSE between predicted knee angles against the ground truth in the unit of degrees.

4.1.4 Baseline Methods

We consider IMUSim and an analytical method as baselines to compare the fidelity of our simulated data because IMUTube also utilizes IMUSim to generate IMU data from 3D global motion trajectories. The analytical method we adopt to compute linear acceleration is Richardson's extrapolation [41], [42]. Compared to taking second-order derivatives, Eq. (5) gives a more accurate estimation with a 4th order error term (as opposed to 2nd order).

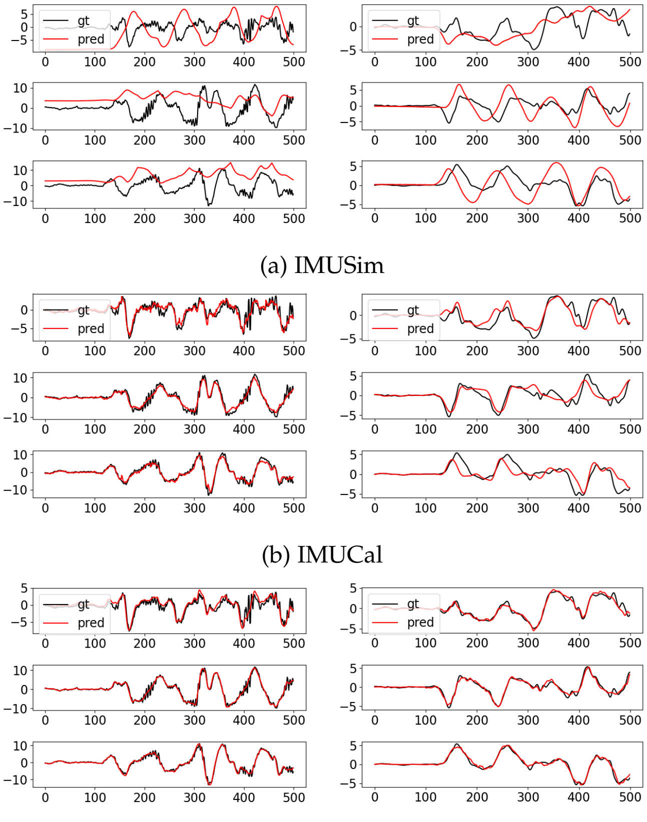
$$acc = \frac{-p(t-2) + 16p(t-1) - 30p(t) + 16p(t+1) - p(t+2)}{12\Delta t^2} \tag{5}$$

The angular velocity of a selected skin region on an SMPL body mesh is calculated by tracking the rotation of its norm vector. The tri-mesh of SMPL model follows the right hand rule, which ensures that the norm vectors of the triangles always point out of the corresponding subject's body. Rotations between consecutive frames are expressed in unit quaternions. Angular velocities in rad/s are computed by multiplying the rotation vector of each frame with the sampling rate. To reduce jitters, we take the average angular velocities of three nearby triangles on the tri-mesh centred in the designated skin region. Lastly, a 4th order ButterWorth low-pass filter is applied to both simulated accelerometer and gyroscope readings for noise reduction [43].

For HAR tasks, we take IMUTube as the baseline, but due to the lack of open source implementations, we include the reported performance on PAMAP2 and Opportunity datasets from [28].

4.2 Fidelity of CROMOSim

In this section, we first provide qualitative and quantitative comparisons between CROMOSim and two baseline methods, namely, the analytical method (IMUCal) and IMUSim in terms of fidelity. We use TotalCapture in this experiment since it contains data from all three required modalities. Two sets of CROMOSim models are trained using MoCap and video data from Subjects 1 – 3 with sensor positions at their right wrist, right foot and pelvis. The models are used to predict accelerometer and gyroscope data on both left and right wrists of Subject 5 from the respective data sources. Next, we analyze the sources of errors in video-based simulations.

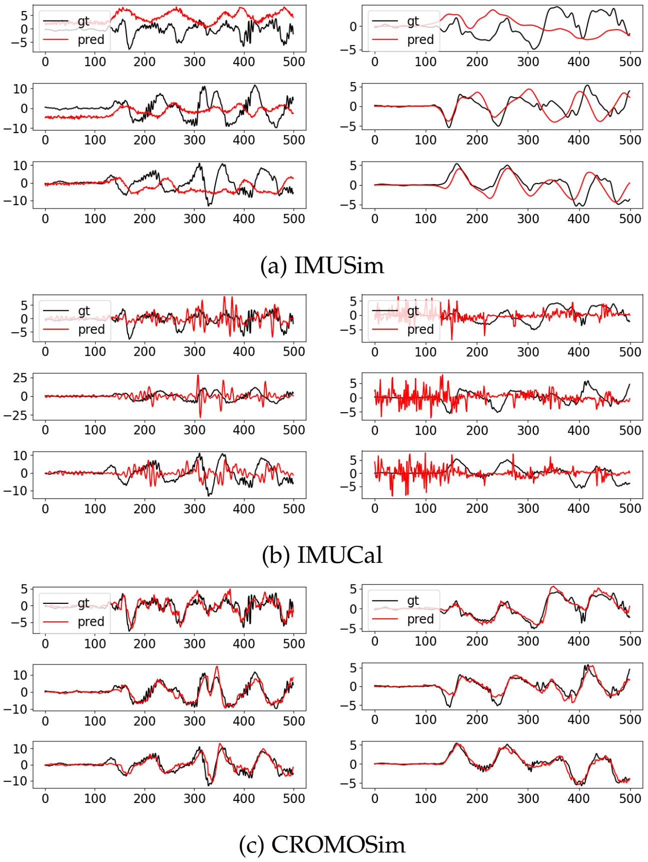


(a) IMUSim

(b) IMUCal

(c) CROMOSim

Fig. 5: Simulated IMU readings on the right wrist of Subject 5 from the MoCap data in TotalCapture. Left: accelerometer data. Right: gyroscope data.



(a) IMUSim

(b) IMUCal

(c) CROMOSim

Fig. 6: Simulated IMU readings on the right wrist of Subject 5 from monocular RGB camera video in TotalCapture. Left: accelerometer data. Right: gyroscope data.

TABLE 1: RMSEs of simulated IMU readings on Subject 5's left wrist across all data trials.

	Acceleration (m/s^2)			Angular velocity (rad/s)		
	IMUSim	IMUCal	CROMOSim	IMUSim	IMUCal	CROMOSim
MoCap extracted SMPL	4.606	1.785	1.602	1.500	1.272	0.801
Video extracted SMPL	6.158	11.824	3.342	1.848	2.578	1.104

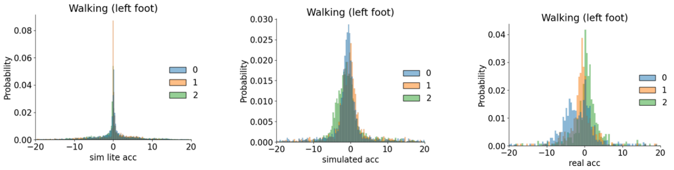
Figures 5 and 6 show the simulated IMU readings from different methods with MoCap and RGB video data, respectively. In these cases, the sensor placement is known but the subject is unseen to the simulator model. From the figures, we observe that the fidelity of IMUSim is low across the board. It is because the default setting of IMUSim filters out too much high-frequency components. IMUCal works well for simulating accelerometer and gyroscope data with MoCap inputs. However, its performance significantly degrades when monocular RGB videos are taken as the source modality. This can be attributed to large noise and relative low accuracy of extracted SMPL body tri-mesh. In contrast, CROMOSim consistently outperforms baseline methods for both data modalities.

4.2.1 *Qualitative and quantitative results*

Table 1 reports the case where both subject and sensor position are unseen to the simulator networks. The quantitative results are consistent with those in qualitative ones shown in Fig. 5 and 6. With MoCap data, the accuracy of

TABLE 2: The analysis of error sources with monucular camera video data.

	VIBE only				Robust CVD	GT global motion
	MPJE (rad)		PVE (m)		PVE (m)	PVE (m)
	RMSE	MAE	RMSE	MAE	RMSE	RMSE
ROM	0.2203	8.3634	0.8099	1.7451	/	/
walking	0.1972	7.9263	1.3741	1.9215	1.1679	0.5046
freestyle	0.2087	8.3627	1.1930	2.1327	0.9607	0.5015



(a) VIBE only (b) Robust CVD (c) Ground truth

Fig. 7: Simulated accelerometer readings on the left foot of Subject 5 from monocular RGB camera video in TotalCapture.

CROMOSim is 187.5% and 11% higher than that of IMUSim and IMUCal for accelerations, respectively, and 87% and 58% for angular velocities. The advantage of CROMOSim is more pronounced with monocular RGB videos, outperforming the next best method (IMUSim) by 84% and 67% for accelerometer and gyroscope data.

4.2.2 Error Analysis

From Table 1, we see that simulated IMU readings from video extracted SMPL have larger errors than those from MoCap. To understand the sources of errors, we conduct further empirical study. Specifically, we analyze the effectiveness of the global trajectory estimation module for root joint, and present the results here. Table 2 summarizes the quality of extracted human pose data on TotalCapture dataset by three approaches, namely, VIBE indicates when the estimation of global trajectory is unavailable, Robust CVD denotes a global motion estimate by the CVD method, while GT global motion refers to align the root node position per video frame with MoCap ground truth. We take the mean per joint error (MPJE, in rad) and per vertex error (PVE, in meters) between the estimated SMPL body mesh from videos and from MoCap data as metrics here. Three types of activities are analyzed: the range of motion sequence (ROM) contains in-place motions with human subjects standing at the center of a laboratory field; the walking sequence involves a person walking around the laboratory; the freestyle sequence corresponds to a freestyle acting and roaming around the room. Clearly, ROM is not affected by global motion trajectory estimations, while the other two are. As the joint angles are extracted by VIBE only, they remain the same with Robust CVD or GT global motion.

From Table 2, there exists a clear gap between the PVE calculated with VIBE only and GT global motion for walking and freestyle, indicating the need to accurately estimate global motion trajectories when motions are not in-place. PVE dropped ∼20% when the Robust CVD is used in video data pre-processing. Fig. 7 shows the probability density function of 3-axis accelerations in a global frame from the two methods in comparison to ground truth. The plots further demonstrate that simulated data are more similar in distribution to the ground truth when global trajectories of the root node are incorporated.

The differences between the estimated global trajectory from Robust CVD and the ground truth can be attributed to two factors. First, we use OpenPose to detect the root node of human subjects in each video frame. OpenPose fails when the resolution is low and the background is complex. Two examples are shown in Fig. 8, where in the left figure a person is running on a trail and in the right figure he is climbing downstairs. Both fail cases are captured from Realworld dataset. The wrongly detected root node will lead to errors in extracted global motion trajectories. Second, calculation of the scale factor is another potential source of errors. To recover real world global motion trajectories from the output of robust CVD, a scale factor is required. In our experiments, it is calculated for 10-seconds video clips. If the human subject in the first video frame is not standing up straight, the scale factor computed using the method in Section 3.3.2 will be larger than the actual values.

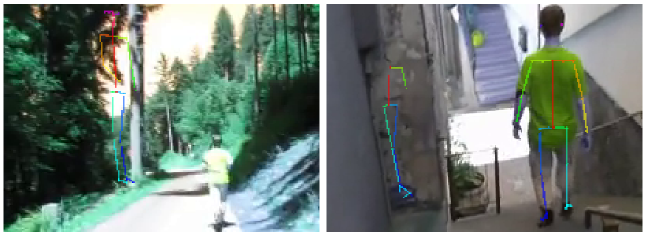


Fig. 8: Typical fail cases of OpenPose in our video data preprocessing, with downscaled video frames, background objects are wrongly recognized as human.

4.3 Applications of CROMOSim in downstream Tasks

4.3.1 HAR Tasks

In this section, we evaluate the utility of CROMOSim in data augmentation for training HAR models. Here we consider three settings: i) R2R, where models are both trained and tested with real IMU data; ii) V2R, where models are trained with simulated data but tested with real data; iii) Mix2R, where models are trained using a mixture of real and simulated data while tested with real data.

We adopt the DeepConvLSTM network proposed in [44] as the task model, while the same simulator neural network trained on the TotalCapture dataset is used here to simulate sensor readings from videos. Evaluations are made on the Realworld, PAMAP2 and Opportunity datasets respectively, with data simulated from the same video source (Realworld dataset). An ablation study was conducted by removing robust CVD from the proposed pipeline, and the resulting approach is called *CROMOSim Lite.* To make the result directly comparable, we followed the experiment protocol in IMUTube [28].

Table 3 reports the average F1 scores of five single-subject-hold out experiments on the RealWorld dataset. Since the authors of IMUTube provide their simulated data on this dataset, we directly replicated their experiments and

TABLE 3: Average and standard deviation of the F1-score of random single subject hold out experiments on the RealWorld dataset. IMUTube* corresponds the scores reported in [28]

	R2R	V2R	Mix2R
IMUTube*	0.730±0.007	0.546±0.008	0.778±0.007
IMUTube	0.729±0.007	0.552±0.005	0.781±0.011
CROMOSim Lite	0.729±0.007	0.580±0.047	0.802 ±0.013
CROMOSim	0.729±0.007	**0.593±0.012**	**0.821±0.003**

TABLE 4: Random single subject hold out evaluation on PAMAP2 dataset with mean F1-score. IMUTube* corresponds to the scores reported in [28]

	R2R	V2R	Mix2R
IMUTube*	0.700±0.016	0.552±0.017	0.702±0.016
CROMOSim Lite	0.702±0.021	0.638±0.009	0.726±0.014
CROMOSim	0.702±0.021	**0.689±0.012**	**0.769±0.009**

TABLE 5: Random single subject hold out evaluation on Opportunity dataset with mean F1-score. IMUTube* corresponds to the scores reported in [28]

	R2R	V2R	Mix2R
IMUTube*	0.887±0.007	0.788±0.010	**0.884±0.007**
CROMOSim Lite	0.862±0.008	0.778±0.013	0.870±0.008
CROMOSim	0.862±0.008	**0.803±0.011**	0.879±0.008

the results are in the second row. For comparison purposes, we also include the scores reported in [28] as the first row. It can be seen the two are quite similar to one another. Even CROMOSim Lite outperforms IMUTube in V2R and Mix2R experiments, while CROMOSim works the best. Moreover, Mix2R achieves much higher F1 scores compared to R2R and V2R, demonstrating the utility of data augmentation with simulated data.

Table 4 and 5 summarize the results from CROMOSim and those reported in [28]. Due to the different sensor placements in the PAMAP2 and the Opportunity datasets, the simulated data provided by the authors of IMUTube cannot be used, so we take their reported performance here. Similar to the RealWorld dataset, CROMOSim outperforms IMUTube for the PAMAP2 datasets but with a more prominent margin; the HAR model trained from Mix2R is still superior to those from R2R and V2R. With the Opportunity data, however, the improvement of Mix2R over R2R is marginal while IMUTube* reports negative results for Mix2R. Although the Mix2R results are lower than those of IMUTube*, the difference is consistent with that for R2R. Therefore, one may consider the two perform comparably for this dataset. The reason for the small benefit of Mix2R in CROMOSim can be attributed to the small number of subjects in Opportunity. With a small number of training subjects, the DeepConvLSTM model does not generalize well to unseen subjects. Despite of the higher level of subject diversity in RealWorld, distribution mapping in IMUTube and CROMOSim in fact forces the distribution of simulated data to be close to the two subjects in the training set. Therefore, the benefit of data augmentation is diminished.

TABLE 6: Average and standard deviation of the F1-score of different domain adaptation methods on the RealWorld dataset, for randomly held-out single subjects.

	V2R	Mix2R
DANN [45]	0.156±0.022	0.488±0.074
ADDA [46]	0.180±0.079	0.607±0.023
CROMOSim	**0.593±0.012**	**0.821±0.003**

To verify the effect of distribution mapping, we have also implemented two unsupervised domain adaptation (UDA) methods, namely, domain adaptive neural network (DANN) [45] and Adversarial discriminative domain adaptation (ADDA) [46] as baseline algorithms. Taking simulated data as the source domain and data collected by real sensors as the target domain, UDA models can be applied to HAR tasks. Table 6 shows the results of the baseline domain adaptation methods against distribution mapping in conjunction with CROMOSim. The UDA methods in V2R and Mix2R scenarios are evaluated by holding out one random subject as the test set. In the V2R case, simulated data from training subjects are combined as the source domain. The 1000 unlabeled real IMU readings per class from each training subject form the target domain. Similarly, in the case of Mix2R, we combine real IMU readings and the simulated ones from training subjects as the source domain, and take the real data from validation subject as the target domain. Each evaluation is repeated 5 times with randomly sampled target domain to report the mean and the standard deviation of F1-score.

From Table 6, it is clear that both UDA methods perform poorly in V2R and achieve much lower F1-scores than distribution mapping. The failure of these two methods can be attributed to the large divergence in IMU data across different subjects. These UDA methods work on a single pair of source and target domains, and assume small distribution gap within the source or target domain. Such assumptions are violated due to subject differences (in V2R) as well as domain gaps between simulated and real data (in Mix2R). Thus, UDA methods failed to learn a feature representation that can properly match the source and target domains. In contrast, distribution mapping is conducted between the data from each subject in the training set (simulated or mixture of real and simulated) and test subject, and thus it is robust to gaps within the source domain.

4.3.2 HPE Tasks

Unlike HAR tasks that are essentially pattern recognition on sensory data, HPE aims to estimate the joint angles of a human body, and requires accurate IMU sensor readings. Therefore, in this section, only MoCap simulated data is utilized.

We have previously designed a DeepBiLSTM network for knee joint estimation. It takes accelerometer and gyroscope readings from sensors on one's thigh and shank to predict 3D knee joint angles. In this set of experiments, We use Subject 1 – 3 in the TotalCapture dataset for HPE model training, and Subject 4's data for validation and real IMU data from Subject 5 for testing. Two sensors (virtual or real) are placed on proximal thigh (ProxTh) and right tibial (RTib) (see Fig. 9). Similar to the HAR tasks, three DeepBiLSTM networks are trained using real data only, virtual data only

TABLE 7: Knee angle estimation. Average RMSE and standard deviation are measured per each axis in degrees.

	X	Y	Z
R2R	15.4550±0.6217	8.3279±0.4751	3.1384±0.0403
V2R	20.8303±1.2644	**7.7459±0.3409**	3.4441±0.1727
Mix2R	**13.9236±0.5875**	8.2440±0.6053	**3.0355±0.2971**

and a mixture of virtual and real data. The size of real data samples from the three training subjects is around 143k, which is 39 minutes long. MoCap simulated data is on the same scale. In R2R and V2R we have 143k real or simulated data for model training, while in Mix2R the training data doubled by mixing the two.

Table 7 summarizes the average RMSEs and standard deviations of 3D knee joint angles in different settings. Note that the RMSEs should be put in the context of range of motions in the TotalCapture dataset, which are $[-11.5220, 152.4866]$, $[-44.3173, 41.3192]$ and $[-17.9953, 30.6022]$ around the x-, y- and z-axes.

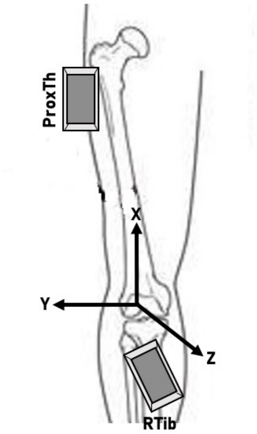


Fig. 9: The sensor placement of knee angle estimation task. Real sensor readings are only available at ProxTh and RTib positions.

From Table 7, we observe that in general Mix2R gives the most accurate estimations followed by R2R. Though the model trained on V2R has lower accuracy in the x-axis, its predictions are comparable to that from R2R in y-axis and z-axis. This phenomenon implies that MoCap generated virtual data using CROMOSim can produce reasonable good HPE models. The observation is consistent with the high fidelity of MoCap simulated data in Section 4.2.

5 Conclusion and Future Work

In this paper, we implemented CROMOSim, a pipeline that simulates accelerometer and gyroscope readings at arbitrary user-designated on-body positions from MoCap and monocular RGB camera videos. A pair of DNN models are trained to learn the functional mapping between imperfect trajectory estimations in a 3D body tri-mesh to IMU data. Experiments showed that CROMOSim can generate higher fidelity data than baseline methods and is useful for downstream HAR and HPE tasks. As part of the future work, we are implementing a graphical user interface and wrapping up CROMOSim as an easy-to-use tool now. Hopefully, it will be open-sourced to the public by this summer. Other directions of further improvements include accelerating the video data processing, proposing a better domain adaption solution to bridge the gap between the distribution of simulated and real data, and experimenting CROMOSim with other data modalities as input such as millimetre wave radar.

Acknowledgement

This project is in part supported by NSERC Discovery and CREATE programs.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[2] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[3] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *arXiv preprint arXiv:2012.13392*, 2020.

[4] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE pervasive computing*, vol. 16, no. 4, pp. 62–74, 2017.

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

[6] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5442–5451.

[7] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[8] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7122–7131.

[9] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5253–5263.

[10] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3d human pose and shape from a single color image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 459–468.

[11] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3383–3393.

[12] A. D. Young, M. J. Ling, and D. K. Arvind, "Imusim: A simulation environment for inertial sensing algorithm design and evaluation," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 2011, pp. 199–210.

[13] H. Ohashi, M. Al-Nasser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, and A. Dengel, "Augmenting wearable sensor data with physical constraint for dnn-based human-action recognition," in *ICML 2017 times series workshop*, 2017, pp. 6–11.

[14] X. Lin, Y. Chen, X.-W. Chang, X. Liu, and X. Wang, "Show: Smart handwriting on watches," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–23, 2018.

[15] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, 2017, pp. 216–220.

[16] O. Steven Eyobu and D. S. Han, "Feature representation and data augmentation for human activity classification based on wearable imu sensor data using a deep lstm neural network," *Sensors*, vol. 18, no. 9, p. 2892, 2018.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[18] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "Sensorygans: An effective generative adversarial framework for sensor-based human activity recognition," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[19] S. Zhang and N. Alshurafa, "Deep generative cross-modal on-body accelerometer data synthesis from videos," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 223–227.

[20] X. Li, J. Luo, and R. Younes, "Activitygan: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 249–254.

[21] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–15, 2018.

[22] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," in *International Conference on Spatial Data and Intelligence*. Springer, 2020, pp. 261–270.

[23] S. Takeda, T. Okita, P. Lago, and S. Inoue, "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*, 2018, pp. 1444–1448.

[24] Y. Liu, S. Zhang, and M. Gowda, "When video meets inertial sensors: Zero-shot domain adaptation for finger motion analytics with inertial sensors," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 2021, pp. 182–194.

[25] V. F. Rey, P. Hevesi, O. Kovalenko, and P. Lukowicz, "Let there be imu data: generating training data for wearable, motion sensor based activity recognition from monocular rgb videos," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 699–708.

[26] V. F. Rey, K. K. Garewal, and P. Lukowicz, "Yet it moves: Learning from generic motions to generate imu data from youtube videos," *arXiv preprint arXiv:2011.11600*, 2020.

[27] V. Fortes Rey, K. K. Garewal, and P. Lukowicz, "Translating videos into synthetic training data for wearable sensor-based activity recognition systems using residual deep convolutional networks," *Applied Sciences*, vol. 11, no. 7, p. 3094, 2021.

[28] H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Ploetz, "Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–29, 2020.

[29] H. Kwon, B. Wang, G. D. Abowd, and T. Plötz, "Approaching the real-world: Supporting activity recognition training with virtual imu data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–32, 2021.

[30] J. Kopf, X. Rong, and J.-B. Huang, "Robust consistent video depth estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[31] J.-H. Mun, M. Jeon, and B.-G. Lee, "Unsupervised learning for depth, ego-motion, and optical flow estimation using coupled consistency conditions," *Sensors*, vol. 19, no. 11, p. 2459, 2019.

[32] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 71–1, 2020.

[33] "Optitrack system," https://optitrack.com/, accessed: 2022-07-09.

[34] "Vicon system," https://www.vicon.com/, accessed: 2022-07-09.

[35] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[36] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019.

[37] W. J. Conover and R. L. Iman, "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, vol. 35, no. 3, pp. 124–129, 1981.

[38] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. P. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors." in *BMVC*, vol. 2, no. 5, 2017, pp. 1–13.

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

[40] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 2010, pp. 233–240.

[41] L. F. Richardson, "The approximate arithmetical solution by finite differences with an application to stresses in masonry dams," *Philosophical Transactions of the Royal Society of America*, vol. 210, pp. 307–357, 1911.

[42] L. F. Richardson and J. A. Gaunt, "Viii. the deferred approach to the limit," *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, vol. 226, no. 636-646, pp. 299–361, 1927.

[43] S. Butterworth *et al.*, "On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

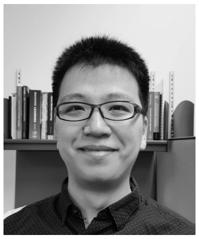
[44] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[45] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[46] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

Yujiao Hao Yujiao Hao received her B.S. degree in Software Engineering and M.S. degree in Computer Science, both from Software Engineering Department of Northeastern University, Shenyang, China. She is currently a Ph.D. candidate in Computing and Software Department of McMaster University, Hamilton, ON, Canada since 2018. Her research interests include sensor-based human activity recognition and motion tracking.

Xijian Lou Xijian Lou received her B.S degree in Biomedical Engineering from Sichuan University, Chengdu, China. She is currently a M.Sc. candidate in Computing and Software Department of McMaster University, Hamilton, ON, Canada since 2020. Her research interests include sensor-based Human Pose Estimation and Deep Learning.



Boyu Wang Boyu Wang received his B.Eng. degree in Electronic Information Engineering from Tianjin University, Tianjin, China, M.Sc. degree in Electrical and Computer Engineering from University of Macau, Macau, China, and Ph.D. in Computer Science from McGill University, Montreal, QC, Canada. He is currently an Assistant Professor with the Department of Computer Science, University of Western Ontario, London, ON, Canada. He is also affiliated with the Brain and Mind Institute and the Vector Institute. He was a Post-Doctoral Research Fellow at the University of Pennsylvania and Princeton University. His research interests include machine learning theory, algorithms, and applications.

Rong Zheng is a Professor in the Dept. of Computing and Software, McMaster University. She is an expert in wireless networking, mobile computing and mobile data analytics. She received the National Science Foundation CAREER Award in 2006, and was a Joseph Ip Distinguished Engineering Fellow from 2015 - 2018. Dr. Zheng has served as the editor of IEEE Transactions on Wireless Communications and IEEE Transactions on Network Science and Engineering. She is currently an editor of IEEE Transactions on Mobile Computing.