


VISUAL RECONSTRUCTION OF HUMAN ACTIVITY DATA GENERATED USING INERTIAL MOTION SENSORS

 **Koustav Barik**
School of Computing
Newcastle University
United Kingdom
k.barik2@newcastle.ac.uk

 **Jacek Cala**
School of Computing
Newcastle University
United Kingdom
jacek.cala@newcastle.ac.uk

August 15, 2024

ABSTRACT

Human Activity Recognition (HAR) has been extensively researched over the past few decades due to its significant potential in applications such as healthcare, sports, smart homes, and security. The ubiquitous nature of smart devices which generate rich streams of motion data has accelerated advancements in this field. HAR research often requires extensive datasets, which are costly and time-consuming to produce. Generative networks can be used to synthetically generate data to address this, but currently, there is no method to visually reconstruct sensor data to validate its consistency with activity labels, ensuring trust in the synthetically generated data. Visual reconstruction helps identify faulty training data and provides a metric to assess the reliability of synthetic data. This study uses motion tracking data to visually reconstruct human movement in three dimensions, with six degrees of freedom, in Unity. We discuss the necessary data preprocessing steps to enhance the quality of training data for generative networks. Gyroscope data is used to derive orientation, while acceleration data is used to derive displacement information. Our approach aids in manually classifying human activities and validating them against a set of ground truth labels, providing a system to improve the consistency and reliability of generative networks and their training data.

Keywords Human Activity Recognition · Visual Reconstruction · Generative Networks · Synthetic Data

1 Introduction

Human Activity Recognition (HAR) has been a much-researched topic over the last couple of decades, holding significant potential for widespread applications in crucial areas such as healthcare, sports, smart homes, and security. With the prevalence of wearable smart devices such as smartphones, smartwatches, and fitness trackers, which generate rich streams of motion data, it has become a rapidly evolving field. These wearable devices are equipped with Inertial Measurement Units (IMUs) or inertial sensors, such as accelerometers, gyroscopes, and magnetometers, which are used to measure the device's acceleration, angular rate, and amplitude of the nearby magnetic field (1).

HAR researchers have developed comprehensive datasets that encompass a diverse range of human activities and demographics, collecting data using the inertial sensors and leveraging multimodal data (2). While a significant number of such datasets exist with labelled data, it is often a laborious process to generate extensive datasets with such diversity. It needs a huge amount of time, money, and effort.

With the maturation of deep learning models and artificial intelligence algorithms like transformers and generative AI, the issue of creating a large scale human activity dataset can be tackled to some extent by synthetically generating the activity data. However, a unique problem remains unaddressed which has limited the progress; the lack of a process to visually reconstruct sensor data and validate its consistency against activity labels with plausible accuracy.

Most work in visual reconstruction of human activity data is done either through real-time data feedback from multiple IMU sensors attached to the subject's body (3) (4), or by combining the real-time sensor data with other sensor data

like cameras (5) (6). Generating a human motion Twin system (7) (8) and character animation using reinforcement learning (9) has also been explored. Although these systems address specific needs within the realm of HAR, they are not well-suited for addressing our current problem.

The ability to visually reconstruct human activity data from sensor readings with promising levels of accuracy holds significant potential for validating synthetically generated data. This validation process would provide a metric to detect bad training data and assess the reliability of the synthetically generated data, enhancing its utility in existing HAR research projects. This research focuses on taking the initial steps to tackle this problem.

The contribution of this research is to use motion tracking data obtained using commodity hardware, like mobile phones and smart watches, to visually reconstruct human movement in three dimensions with six degrees of freedom, with the intention to manually classify human activity and validate it against a set of ground truth labels. If successful, it could greatly aid in data collection and validation of both real and synthetic data. By leveraging Unity, a powerful and flexible game development platform, this research includes developing a novel system to enhance the interpretability of accelerometer and gyroscope data through visual representation, mimicking real-life movements. This visual approach aids in the recognition of activities, finding issues with the data, and offers an engaging way for users and researchers to understand the data.

This paper is organized as follows. Section 2 briefly refers to some of the background material needed to understand the research problem. Section 3 describes our proposed method for visually reconstructing the activities from sensor data. Section 4 shows the results of reconstructing the dataset visually and evaluating against the ground truth labels. The next section concludes the contribution of this paper to the HAR applications followed by plans and suggestions for future work.

2 Background

HAR has been a focal point of research for several decades, with numerous studies exploring different methodologies and datasets to improve the accuracy and robustness of activity recognition systems. Due to the ubiquitous nature of smartphones and wearable smart bands, it has become an increasingly important research area (10). This section reviews the existing literature on HAR, focusing on the topic of using IMU data, the development of datasets, synthetic data generation, and visual reconstruction techniques.

2.1 Human Activity Recognition Using IMU Data

The utilization of inertial measurement unit data for HAR has been extensively studied. As a starting point, we went through some of the surveys (11) and repositories (2) listing the trends, problems, and state-of-the-art research in HAR, which helped us narrow down the relevant areas concerning our study. Early work by Bao and Intille (12) demonstrated the feasibility of using accelerometers placed on different parts of the body to recognize activities like walking, running, standing and sitting. Subsequent research by Anguita et al. (13) further advanced the field by leveraging smartphone accelerometers to classify activities using machine learning algorithms. These studies highlighted the potential of accelerometer data in providing accurate and non-invasive monitoring of human activities. Incel et al. (14) provided an extensive survey on activity recognition using mobile phone sensors including a taxonomy of existing work especially focusing on the issues of health and well-being. Zhipeng et al. (1) presented a comprehensive survey on IMU-tracking techniques on mobile and wearable devices, and revealed the key challenges, and the possible directions to address these challenges in IMU-based motion tracking. Kok et al. (15) talked about the issues encountered and their potential fixes in using inertial sensors for position and orientation estimation. We involve brief discussions about the challenges relevant to our study.

2.1.1 Identifying the reference frame

For motion tracking, we need to define the reference frames clearly. 1. The body frame is the frame defined by the device using which data is collected, e.g., sensor readings from smartphones. The navigation frame is the global and absolute frame. For inertial motion tracking on mobile and wearable devices we typically consider these two frames, also known as the Local Reference Frame (LRF) and Global Reference Frame (GRF) (16). The GRF is defined as $\langle North, East, Up \rangle$. So, if an accelerometer has readings on the X-axis, then this acceleration is along the device's X-axis, not the X-axis in the global framework. Therefore, we need to transfer the accelerometer and gyroscope readings from the LRF to the GRF to obtain correct tracking results.

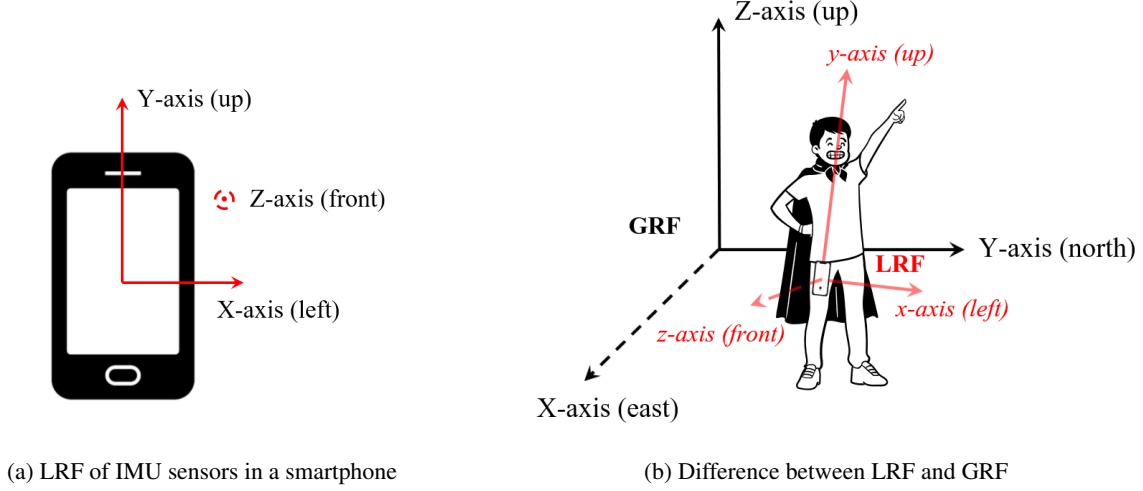


Figure 1: Explanation of Local and Global Reference Frames

2.1.2 Calculating the orientation

Position and orientation are not directly measured by an IMU. Instead, the time derivatives angular velocity and acceleration are measured. The initial orientation of a device refers to its facing direction relative to the GRF, which is crucial for determining the rotation matrix that maps the LRF to the GRF (1). Establishing the initial orientation requires the identification of global anchors, which are reference measurements that remain constant in the GRF, regardless of changes in the LRF. Gravity, as measured by the accelerometer, can act as a reliable global anchor. However, it is likely to be subject to high levels of noise; accelerations due to motion will corrupt the measured direction of gravity. As the device rotates, its orientation changes over time, necessitating the determination of both the initial orientation and the subsequent changes in orientation between consecutive samples. A gyroscope measures angular velocity which, if initial conditions are known, may be integrated over time to compute the sensor's orientation. This brings us to the Madgwick's orientation estimation algorithm (17), whose task is to compute a single estimate of orientation through the optimal fusion of gyroscope and accelerometer measurements.

The most common form of representing orientation is via euler angles. They are conceptually relatable while picturing a 3D space. But using euler angles might result in encountering an issue known as the gimbal lock that prevents certain rotations when two axes align. A widely adopted method for representing orientation that helps solve this problem is through the use of unit quaternions. The algorithm makes extensive use of quaternion mathematics. Quaternions (18) are widely used in many orientation estimation algorithms, e.g. (19) (20). A quaternion is a four-dimensional complex number defined as $q = w + xi + yj + zk$, where w, x, y, z are real numbers, and i, j, k are the fundamental quaternion units satisfying $i^2 = j^2 = k^2 = ijk = -1$. A unit quaternion is a quaternion where the norm $\|q\| = \sqrt{w^2 + x^2 + y^2 + z^2}$ equals 1. Unit quaternions are particularly useful in representing rotations in three-dimensional space. The conjugate of a quaternion q is denoted by q^* and is given by $q^* = w - xi - yj - zk$. The product of a quaternion with its conjugate results in the square of its norm, i.e., $q \cdot q^* = \|q\|^2$. For unit quaternions, since $\|q\| = 1$, it follows that $q \cdot q^* = 1$. Quaternion multiplication is non-commutative, meaning the order of multiplication matters, and it is defined by the distributive property along with the relationships $ij = k$, $jk = i$, and $ki = j$, while reversing the order negates the result, such as $ji = -k$.

The core principle of the algorithm (17) involves integrating gyroscope data to calculate the estimated orientation while introducing a feedback term to correct any drift. The drift can occur due to accumulation of the cumulative errors over time causing deviation of the estimated orientation from the true orientation. The feedback term is calculated as the error between the current measurement of orientation provided by the algorithm output and the instantaneous measurement of orientation indicated by the accelerometer, which is improved using a gradient descent algorithm and multiplied by a gain factor. This gain factor determines the balance between trusting the gyroscope and incorporating corrections from other sensors, allowing the algorithm to function as a complementary filter. A higher gain increases reliance on the accelerometer, which helps mitigate drift but may introduce errors during periods of linear acceleration. A lower gain just trusts the gyroscope. The filter continuously updates the sensor's orientation by integrating the estimated rate of orientation change, accounting for the sensor's motion and any sensor noise.

2.2 Shortlisting of HAR Datasets

Comprehensive datasets are crucial for training and evaluating HAR systems. Before smartphones became popular, external sensors and sensors embedded in wearable garments or straps were used to create HAR datasets. Some recent datasets were also created using wearable sensors. With the availability and popularity of smartphones, more HAR datasets based on its embedded accelerometer or gyroscope are available. We focused on publicly available datasets with more than 30 subjects so that synthetic data generation could be supported. Anguita et al. (13) developed the UCI HAR dataset by collecting accelerometer data from 30 participants. Vavoulas et al. (21) created the MobiAct dataset, which includes accelerometer data from 57 participants. Similarly, Micucci et al. (22) introduced the UniMib dataset, based on accelerometer data from 30 participants. Weiss et al. (23) developed the WISDM dataset, incorporating accelerometer and gyroscope data from 51 participants using both smartphones and smartwatches. Karas et al. (24) created a PhysioNet (25) HAR dataset capturing accelerometer data from 32 adult participants using ActiGraph GT3X+ devices. These datasets have been instrumental in benchmarking HAR algorithms and facilitating the development of more robust models. Table 1 shows the comparison of specifications for these five public datasets.

Dataset	Year	Subjects	No. of Activities	Activity Duration (s)	Freq. (Hz)	Raw Data	SmartPhone	
							Acc	Gyro
WISDM	2018	51	18	180	20	Y	Y	Y
UCI HAR	2012	30	6	30-36	50	N	Y	N
MobiAct	2016	57	9	10-300	20	Y	Y	N
UniMiB	2016	30	9	15-30	50	Y	Y	N
PhysioNet	2021	32	5	15-120	100	Y	Y	N

Table 1: Comparison of Public Datasets

2.3 Synthetic Data Generation for HAR

To address the challenges of dataset creation, recent studies have explored the use of synthetic data generation. Li et al. (26) employed generative adversarial networks (GANs) to create synthetic accelerometer data for HAR using HASC2011corpus dataset which is publicly available. Chen et al. (27) explored data augmentation on the UCI HAR dataset (13). These works demonstrated that synthetic data could augment existing datasets, improving the performance of HAR models. However, the validation of synthetic data remains a challenge, as there are few established methods to ensure its accuracy and consistency with real-world activities.

2.4 Visual Reconstruction Techniques

Visual reconstruction of sensor data have been explored to find tools and techniques that help with intuitive understanding and validation of human activities. Several works as listed in Desmarais et al. (28) and Nyugen et al. (29) have utilized pose estimation approaches to visualize human movements, combining IMU data with camera feeds to create detailed 3D reconstructions. Van Wouwe et al. (30) uses Opensense by Opensim (31) for their pose estimation approach. Further, there is the concept of digital twins where the human motion is replicated in real-time for applications and advancements in fields of healthcare, industry, aerospace, sports and many more. Sun et al. (7) discusses extensively about advancements in healthcare. Zhou et al. (8) talks about human centric applications in general. Pin Ge et al. (32) and Tahmid et al. (9) uses Unity game engine (33) for their human motion twin approach. While effective, the above methods often require real-time multi-sensor setups that need rigorous calibration and are not easily scalable. Moreover, it doesn't translate well for our problem in hand where we look to visually reconstruct motion data of an existing dataset.

3 Methodology

This section outlines the methodology adopted for the visual reconstruction of human activity recognition (HAR) data obtained from inertial motion sensors. The overall process for our study is shown in Figure 2.

The methodology section is divided into the following subsections: dataset selection, framework selection, data preparation, and visual reconstruction methods.

3.1 Dataset Selection

We select the WISDM dataset as the primary dataset for our research due to several compelling reasons that can be seen in 1. The WISDM dataset stands out due to its significant amount of data both in terms of the number of subjects and

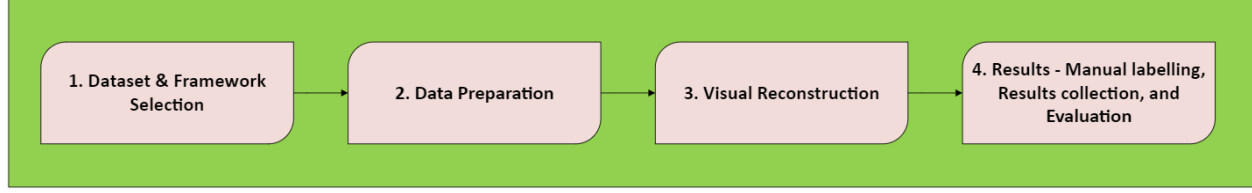


Figure 2: Flow Diagram of the steps included in this study.

the variety of activities recorded for each subject. With data from 51 subjects and 18 activities, it provides a substantial amount of data that supports the training of generative networks to produce synthetic data.

Moreover, one of the highlighted strengths of the WISDM dataset is its provision of raw data. This raw data allows us to identify and correct inconsistencies in the recordings, ensuring the integrity of our analysis. Additionally, having access to unprocessed data enables us to develop customized preprocessing and manipulation methods tailored specifically to our research needs.

Additionally, the dataset includes recordings from both smartphones and smartwatches, encompassing various devices with embedded accelerometers and gyroscopes. In our study, we utilize data exclusively from smartphone sensors. We choose this approach because reconstructing data from sensors placed in a single location on the subject’s body is considered a fundamental initial step in visual reconstruction. Future studies could expand on this work by incorporating data from both smartphones and smartwatches.

The WISDM dataset contains data from 51 subjects coded 1600 through 1650, with 18 activities coded from A through S. We choose data only for dynamic activities, namely, Walking (A), Jogging (B) and Stairs (C), recorded by the smartphone for this study. We specifically choose these activities since they exhibit periodicity. Moreover, they involve the movement of the full frame of the subject’s body. Even if the smartphone is kept in the right hip pocket, the full body movement associated with these activities results in distinctive and recognizable patterns in the sensor data, making the visual reconstruction relatively easy to distinguish.

3.2 Framework Selection

We examine various frameworks based on their fit and limitations with respect to our study, leading to several challenges and insights, as detailed below.

- The Quaternion Motion Tracking (QMT) python toolkit (34) is a framework with a collection of functions, algorithms, visualization tools, and other utilities with a focus on IMU-based motion tracking. It aids efficient processing and analysis of motion data obtained from inertial measurement units (IMUs). Although some web applications are available for modeling the data, challenges arise when attempting to replicate the examples from the project’s GitHub repository on a local machine. Existing functions give errors when trying to run the basic examples.
- MuJoCo (Multi-Joint dynamics with Contact) (35) is a physics engine designed primarily for simulating and controlling complex robotic systems and articulated bodies using sensor and actuator data. While MuJoCo provides some fundamental models that could aid in visualizing the motion data, their operational functionality necessitates rotational data at the joints to accommodate torque as input. This requirement poses a limitation to our exploration of the framework since we only have accelerometer and gyroscope data from smartphone kept in subject’s pocket.
- OpenSense (31) by OpenSim serves as a specialized tool tailored for motion capture and analysis tasks. It is proficient in handling data sourced from wearable sensors and inertial measurement units (IMUs), enabling users to derive a diverse array of biomechanical metrics and delve into the analysis of human movement patterns. Integration of a pre-existing model with example data provided by OpenSense gives promising visual outputs, suggesting compatibility. However, the input data format required to feed a model requires accelerometer and quaternion data for multiple joints and specific file formats which limits the use of this framework for our study.
- The Unity game engine (33) provides the best fit for visually reconstructing human motion data. We see that we can use both accelerometer and gyroscope data to model the movement of a human being with the help of some data preprocessing.

3.3 Data Preparation

Data preparation is a critical step in ensuring the quality and reliability of the data to be used for visual reconstruction and more importantly to train a generative network to produce synthetic data. Our initial accelerometer data includes columns: Subject-id, Activity, Timestamp, accel_x, accel_y, accel_z and our initial gyroscope data includes columns: Subject-id, Activity, Timestamp, gyro_x, gyro_y, gyro_z.

Our data preparation process involves an iterative approach of data understanding and data preprocessing which is outlined in Figure 3.

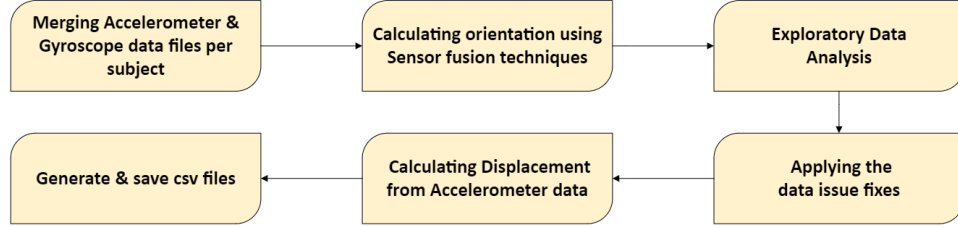


Figure 3: Data Preparation

We first select the required activities by loading all the data files and create separate dataframes for accelerometer and gyroscope data. We see that the total number of sample points for the smartphone accelerometer is significantly higher (1338016) than the total number of sample points for the smartphone gyroscope (1006698). We merge the two dataframes based on the DateTime column, derived from the unix Timestamp column, which ensures no common data is left out.

To compute the device orientation from accelerometer and gyroscope data, we use the *imufusion* Python library (36), which implements the improved Madgwick’s algorithm discussed in 2.1.2. The algorithm functions as a complementary filter, blending high-pass filtered gyroscope measurements with low-pass filtered accelerometer data. For our implementation, we utilize the quaternion to euler function to convert the quaternion output of the algorithm into euler angles, facilitating the analysis of orientation dynamics in our study. Since movements during human activities seldom result in a complete 180° axial rotations of the smartphone, gimbal lock can be overlooked over the short duration of the activities.

Once we have the euler angles added in our dataframe, we do an exploratory data analysis for each subject categorised by each activity. Our examination leads us to discover some major issues with the data. The data issues and their fixes are discussed below:

- The orientation of the smartphone is inconsistent across different activities for most subjects. This inconsistency could stem from placing the smartphone in the subject’s pants pocket with an incorrect orientation or from the smartphone’s position being altered during the activity.

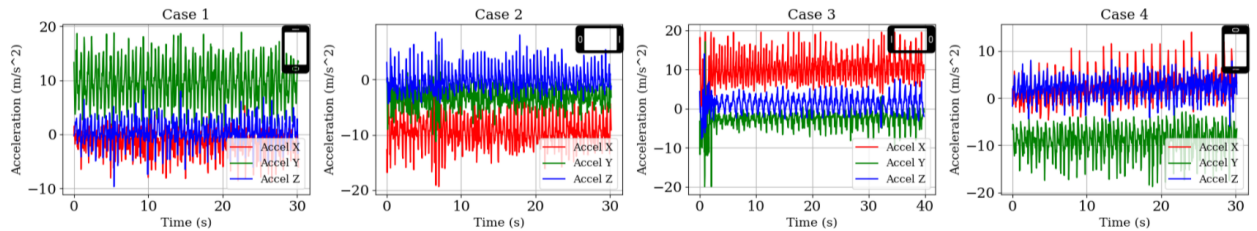


Figure 4: Example of Orientation Issues in the Accelerometer Signals

Figure 4 demonstrates the recorded smartphone accelerometer signals from four different subjects during the walking activity, each subject having the smartphone in a different orientation. We label these four orientations as cases 1 through 4. In case 1, the Y-axis of the accelerometer is parallel to and aligned with gravity. Conversely, in case 4, the Y-axis is parallel to but opposite gravity. In case 3, the X-axis is parallel to and aligned with gravity, whereas, in case 2, the X-axis is parallel to but opposite gravity. We fix this by shifting the signals, i.e., adding two times the absolute average of the amplitude and exchange X and Y axes if the average amplitude of the X-axis was more significant than the average amplitude of the Y-axis using (37)

as reference. Figure 5 shows the signals post implementation of orientation fix. As a quick verification, we can confirm that the gravity is acting in the y -axis for all the signals.

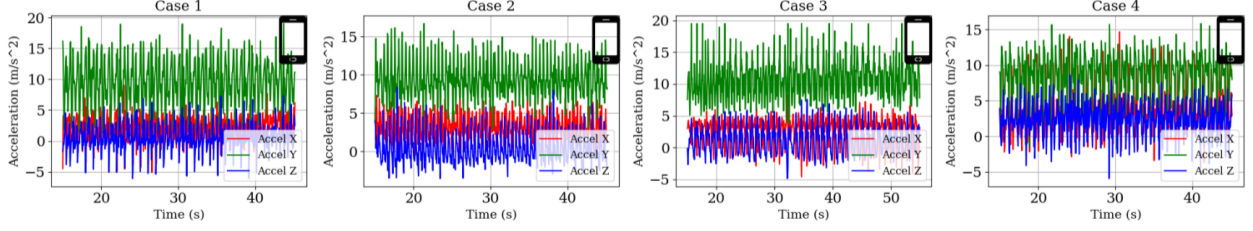


Figure 5: Accelerometer Signals post orientation fix

- We see that the length of the activities vary for different subjects. For example, the number of sample points for one activity for some subjects is almost 3600 points (e.g., subjects 1600, 1602, 1604), 4500 points (e.g., subjects 1601, 1647), 14280 points (e.g., subject 1629). This can be attributed to the fact that multiple devices were used to record the data which had different sampling frequency rates. We fix this by resampling the frequency to 20 Hz using (37) as reference.
- For some activities, in case of some subjects, there is not enough data. For e.g., in case of Jogging, subjects 1614, 1641, 1642, 1644, 1645, 1646, have less than 300 data points. We fix this by excluding data for such subjects from our study.
- We observed that the initial seconds of the signals captured transition actions (e.g., standing up then starting to walk) before the activity began. To eliminate these non-activity-related segments, we removed the first 15 seconds of the recordings.

Having fixed the data issues, we proceed to calculate displacement from acceleration using (38) as reference. We first calculate the velocity from acceleration data. Figure 6 represents an abstraction of data yielded from the accelerometer.

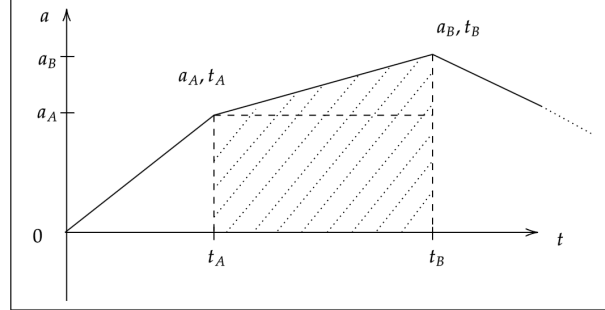


Figure 6: Visual abstraction of sample Accelerometer data

Velocity Δ_v at any time interval Δ_t is the integral of acceleration, which can be calculated as the area under the curve denoted by the shaded region in the figure. We find the area of the trapezoid by summing the area of the rectangle and the triangle.

$$\begin{aligned}
 v_B - v_A &= a_A (t_B - t_A) + \frac{1}{2} (t_B - t_A) (a_B - a_A) \\
 &= a_A t_B - a_A t_A + \frac{1}{2} (a_B t_B - a_A t_B - a_B t_A + a_A t_A) \\
 &= \frac{1}{2} t_B (a_A + a_B) - \frac{1}{2} t_A (a_A + a_B) \\
 &= \frac{1}{2} (a_A + a_B) (t_B - t_A) \\
 \Delta_v &= \frac{1}{2} (a_A + a_B) \Delta_t
 \end{aligned} \tag{1}$$

We modify this equation while calculating velocity for the axis in which gravity is acting for any particular activity, i.e., the y -axis in the case of walking, jogging, and stairs, after applying the necessary orientation fixes. In our case, we need

to subtract the amount $1g = 9.8m/s^2$ to take the force of gravity on Earth into account, since the WISDM data was collected with this value of $1g$.

$$\Delta_v = \frac{1}{2} ((a_A - 9.8) + (a_B - 9.8)) \Delta_t \quad (2)$$

Similarly, to calculate displacement from velocity:

$$\Delta_d = \frac{1}{2} (v_A + v_B) \Delta_t \quad (3)$$

We use the above formulas to derive instantaneous velocity and displacement values for each data point. Figure 7 shows examples of accelerometer signals and the derived velocity and displacement signals for the activities for subject 1600. It is important to note that the mean of the velocity and displacement signals for the x and z axes are non zero implying that there might be some signal errors that result in this non-stationary nature of the signal.

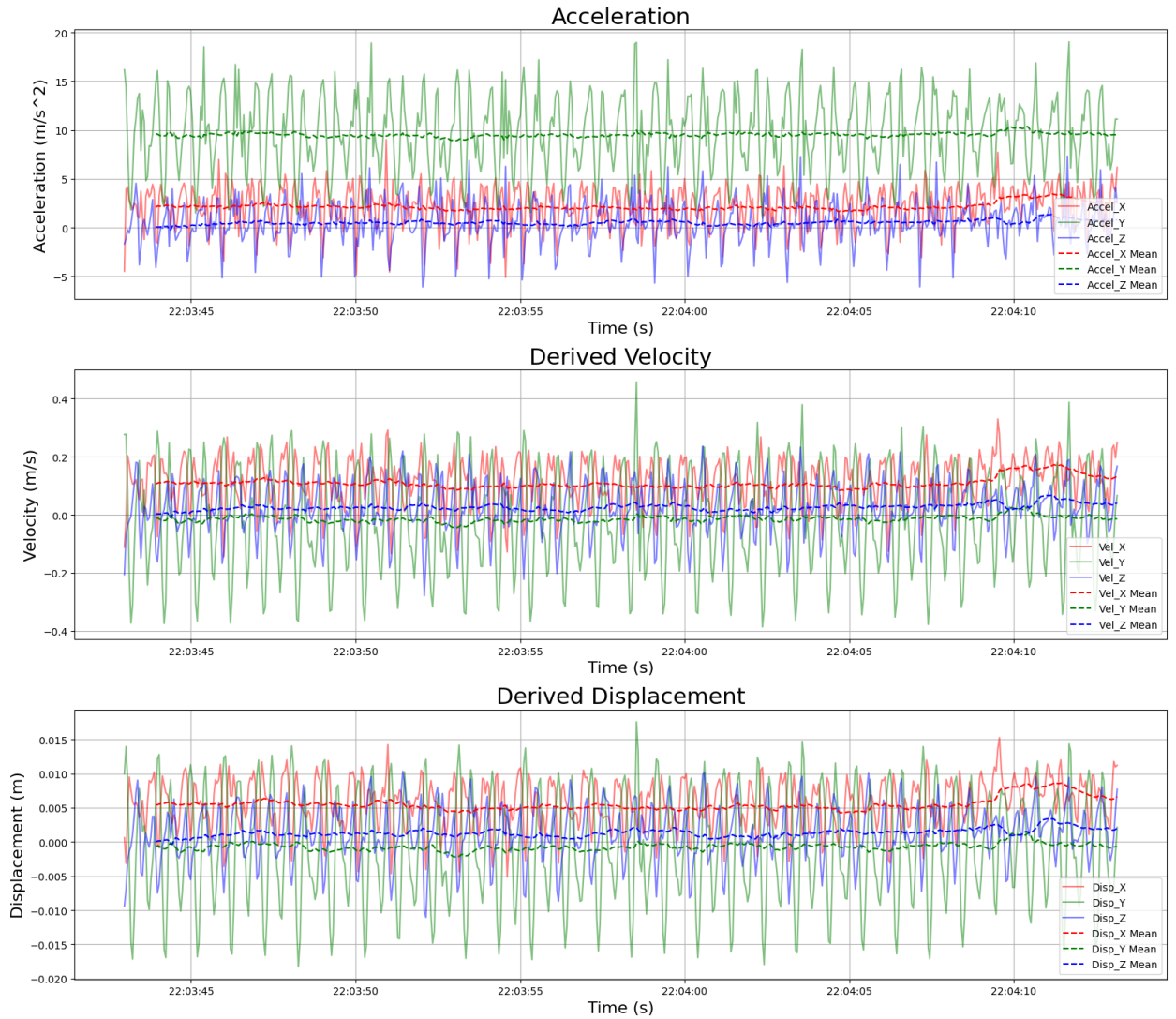


Figure 7: Examples of derived velocity and displacement data from acceleration data for an instance of walking activity, with signal means calculated using a rolling window of 20

Finally we generate CSV files for each activity per subject to input into Unity as a part of the visual reconstruction process. The final CSV data includes columns: Subject-id_accel, Activity_accel, Timestamp_accel, accel_x,

accel_y, accel_z, gyro_x, gyro_y, gyro_z, accel_magnitude, gyro_magnitude, Roll, Pitch, Yaw, vx, vy, vz, dx, dy, dz, Time_diff

3.4 Visual Reconstruction Methods

In this study, we utilize the Unity game engine to visually reconstruct displacement and orientation data from human motion activities. The process involves several key steps to accurately replicate the recorded movements and rotations within a simulated environment outlined in Figure 8. This approach provides a comprehensive visualization of motion data, facilitating better analysis and understanding of human activities.

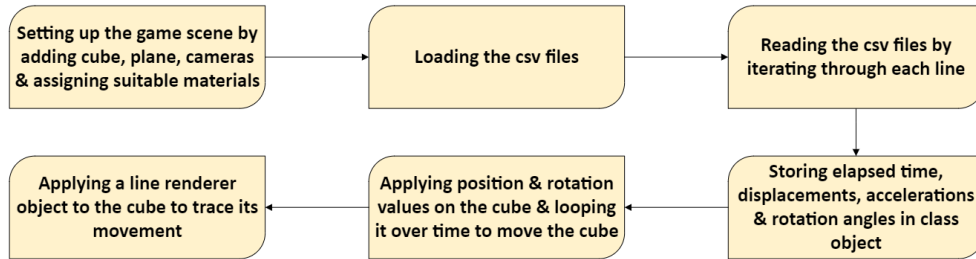


Figure 8: Visual Reconstruction

The first step in our implementation is setting up the game scene within Unity. We begin by adding essential components such as a cube, which serves as the primary object representing the human motion, and a plane that acts as the ground. The cube dimensions and positioning with respect to the plane is done keeping in mind that it depicts a smartphone kept in the hip pocket of a subject. Multiple cameras are strategically placed to capture various perspectives of the cube’s movement. Suitable materials are assigned to the objects to enhance visual clarity and realism.

Next, we focus on loading and reading the CSV files containing the motion data. These files include detailed records of elapsed time, displacements, accelerations, and rotation angles. By iterating through each line of the CSV files, we extract the relevant data and store it within a class object designed to hold these values. This object-oriented approach ensures efficient management and accessibility of the data throughout the simulation. Once the data is appropriately stored, we apply the position and rotation values to the cube to animate its movement. By looping over time, the cube’s position and orientation are continuously updated based on the recorded displacements and rotation angles. We use *WaitForSeconds* function in conjunction with the recorded elapsed time to ensure that the updates to the cube’s position and rotation happen at consistent intervals which is roughly around 0.05 seconds, with the frequency being 20 Hz, making it 20 observations in a second. The function makes Unity’s coroutine wait for the specified number of seconds, accurately reflecting the real-world timing of the recorded movements. This is crucial for maintaining the fidelity of the visual reconstruction of the sensor data. This dynamic application of data allows us to accurately replicate human motion as captured by the sensors. The pseudo-code for the above process is included below in Algorithm 1.

To further enhance the visualization, we incorporate a line renderer object attached to the cube. This line renderer traces the movement of the cube, leaving behind a visible trail that depicts the path taken during the simulation. The trail provides a clear and continuous representation of the motion, making it easier to analyze the trajectory and identify patterns or irregularities.

4 Experimental Results

4.1 Visual Reconstruction

Figure 9 presents examples of the visually reconstructed activities of Walking, Jogging, and Stairs, as generated using our proposed system in Unity. It is important to note that visual reconstruction identification is most effective when observed through real-time videos (39).

The visual reconstruction of displacement and orientation data provides valuable insights into the different human motion activities. For the activity of Jogging, the cube representing the phone placed in the subject’s hip pocket exhibits more frequent vertical oscillations compared to the activities of Walking and Stairs. This indicates a higher frequency of ups and downs during Jogging, which is consistent with the nature of this activity. The frequency of oscillations for Walking and Stairs is almost similar. The Stairs activity reveals a distinct pattern characterized by changes in pace and orientation at regular intervals. These changes correspond to the landing areas encountered after each flight of

Algorithm 1 Displacement Data Handler

```
1: procedure DISPLACEMENTDATAHANDLER
2:   Attributes:
3:     fileName  $\leftarrow$  "Assets/filename.csv"
4:     elapsedTimeColumnName  $\leftarrow$  "Time_diff"
5:     <displacementDataColumnName>  $\leftarrow$  "dx", "dy", "dz"
6:     <orientationDataColumnName>  $\leftarrow$  "Roll", "Pitch", "Yaw"
7:     displacementData  $\leftarrow$  List of DisplacementEntry object type
8:     currentIndex  $\leftarrow$  0
9:     modelTransform  $\leftarrow$  Transform

10:  Class: DisplacementEntry
11:    Attributes:
12:      ElapsedTime  $\leftarrow$  Float
13:      Displacement  $\leftarrow$  Vector3
14:      Roll, Pitch, Yaw  $\leftarrow$  Float
15:    Constructor: Initializes attributes

16:  procedure START
17:    Initialize displacementData
18:    Load CSV data
19:    Start playback coroutine

20:  procedure LOADCSV
21:    Open and read CSV file
22:    Parse header for column indices
23:    while not end of file do
24:      Read line and split into values
25:      Extract and parse values
26:      Create DisplacementEntry object
27:      Add to displacementData

28:  procedure PLAYBACKDATA
29:    while true do
30:      if currentIndex < displacementData.Count - 1 then
31:        currentData  $\leftarrow$  displacementData[currentIndex]
32:        nextData  $\leftarrow$  displacementData[currentIndex + 1]
33:        Update modelTransform.position using currentData.Displacement
34:        Update modelTransform.rotation using currentData.Roll, Pitch, Yaw
35:        Calculate waitTime from difference between nextData.ElapsedTime and currentData.ElapsedTime
36:        currentIndex ++
37:        WaitForSeconds(waitTime)
38:      else
39:        break
```

stairs. The visual data shows periodic pauses and shifts in orientation, reflecting the subject's adjustment of position and direction on these landings before continuing with the next flight of stairs.

These results highlight the effectiveness of using a visual reconstruction approach to analyze and differentiate between various human motion activities, providing a clear and detailed understanding of the distinct movement patterns associated with each activity.

4.2 Quantitative Results

Across the 50 subjects in each activity category, the visual reconstruction results were not plausible for all subjects. Walking showed accepted levels of reconstructions for 28 subjects, Jogging for 17 subjects and Stairs 18 subjects. There were cases where the reconstructions showed exaggerated sinking effect - 19 for Stairs, 18 for Jogging and 7 for Walking. Rest of the subjects were either dropped due to less data or did not have any data present in the dataset.

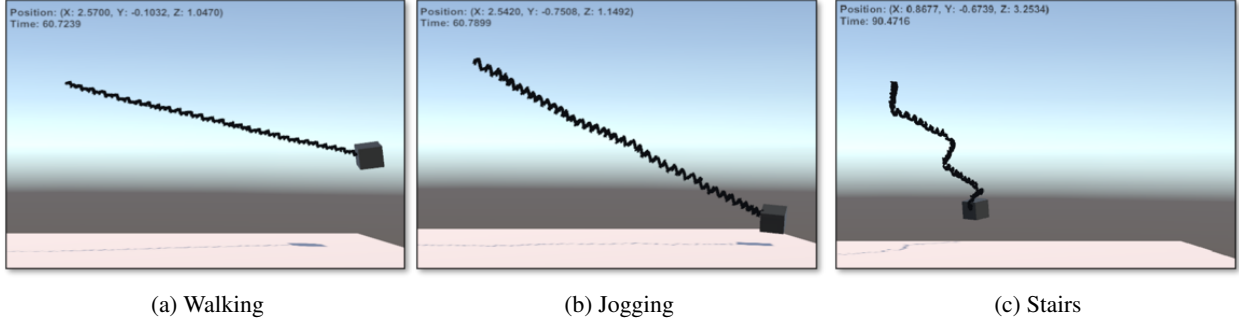


Figure 9: Examples of visually reconstructed activities

The reconstructed activities were visually inspected and compared with the ground truth labels to assess their plausibility and consistency with the expected movement patterns (39). This involved subjective evaluation by human observers familiar with the activities.

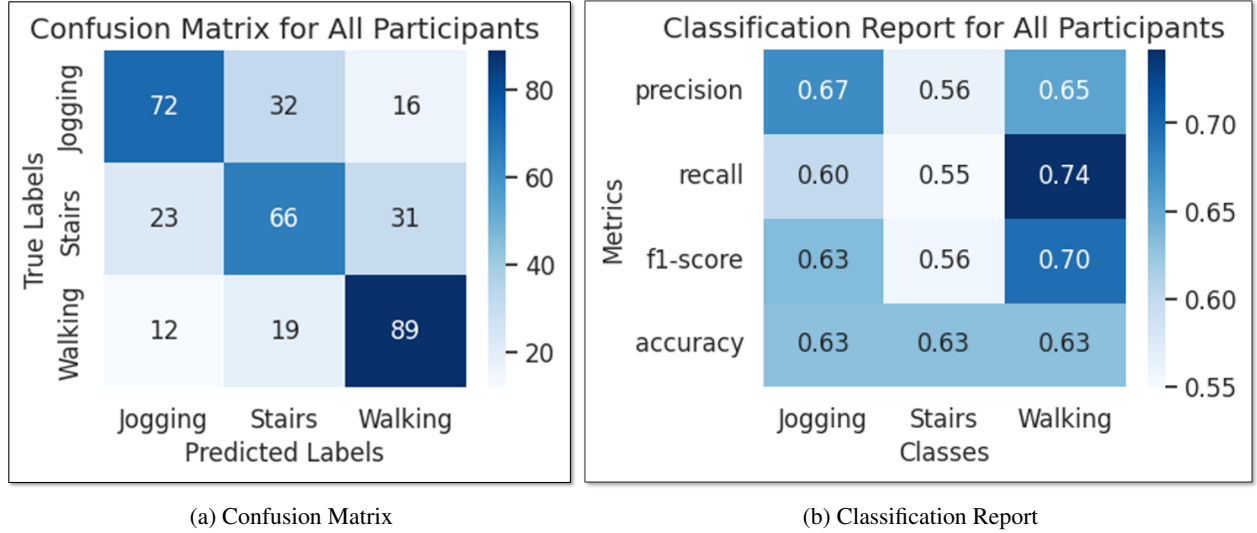


Figure 10: Evaluation Metrics

To mitigate potential bias in the evaluation performed solely by the author, we conducted a survey incorporating unlabelled cases. It included a total of 15 activities, composed of five sets of the three distinct activities, which were randomly selected and included in the survey. 24 participants were provided with some contextual information regarding the study to aid in their assessment. This approach ensures a more objective and comprehensive evaluation of the activity recognition system, leveraging diverse perspectives to validate the results. The results of the survey are shown in Figure 10a. Metrics such as accuracy, precision, recall, and F1-score were calculated as part of a classification report to quantify the alignment between the predicted reconstructed activities and the ground truth labels as shown in 10b.

The classification process demonstrated varying levels of accuracy across different activities. Specifically, the percentage of activities predicted correctly was 74% for Walking, 60% for Jogging, and 55% for Stairs. Among the activities reconstructed, Walking is the easiest to identify. This is because Walking movements are distinct and uniform, making them more recognizable in the visuals. In some cases though, the reconstruction of Stairs and Walking activities appears nearly identical due to their similarity in the frequency of the smartphone moving up and down, making it challenging to differentiate between them. Moreover, Jogging often got confused with Stairs too, since Jogging showed sinking effect which made it look like downstairs Walking activity.

Overall the system achieved an accuracy of 63%, indicating a moderate level of performance across the different activities. These results demonstrate the system's strengths in identifying walking and jogging activities, while suggesting room for improvement in the classification of stair-related activities.

4.3 Evaluation

The results of the proposed system for visually reconstructing human activities reveals that the outputs can be identified by humans with a reasonable degree of accuracy. The reconstruction of some activities is inconsistent, as in some cases, the reconstructed movements do not follow a straight line as expected. These deviations make it harder to accurately identify the activity compared to the more consistent representations of them. These inconsistencies in the reconstructions highlight the need for further refinement in the data preparation process to enhance the overall accuracy of the visual reconstruction system. Our study revealed certain limitations:

- A notable issue was the sinking effect observed along the Y-axis in the reconstructions. It is likely to be caused due to the accumulation of errors in the data preparation phase. During the conversion of acceleration to displacement, we assumed the gravity to be always working in the y -axis of the smartphone. In reality it is not always the case, since the smartphone keeps changing orientation during a subject's movement. Different types of sensor noises, like time varying inherent and environmental sensor noise (1), present during the data collection process can also play a part in this issue which needs to be investigated. The same noises could also be responsible for the non-zero mean of the acceleration signals.
- Furthermore, the interpretability of the visually reconstructed activities was not consistent across all subjects for specific activities. The small test size also poses a risk that the accuracy of our findings might diminish when applied to a larger test set. We also saw during our result collection phase that comparing the reconstructed visuals against each other and labelling them proves to be easier than assessing isolated visuals when there is no contextual background.
- Another significant limitation is the current system's inadequacy in handling multi-sensor data obtained from sensors placed at multiple body locations on a human subject. This constraint highlights the need for further research and development to enhance the robustness and applicability of our visual reconstruction system in more complex scenarios. A better grasp of the Unity game engine is also required to execute the above idea.

5 Conclusion

Our study identified that we need unscaled raw data for accurate visual reconstruction. Further, we found that it's essential to check and fix the data before using it to train generative networks and produce synthetic data. These checks involved ensuring that the smartphone or sensor orientation is consistent, making sure the sensor sampling frequency is uniform, and removing any bad data, such as readings with none or very little data. We employed sensor fusion technique that enabled us to convert gyroscope data collected in the local reference frame of the smartphone to orientation data in the form of euler angles in the global reference frame. Additionally, we used acceleration data to derive displacement data, getting positional information of the activity movements in the process.

Regarding visual reconstruction, we found that the reconstructions produced by our framework are detailed enough to be understood by viewers to a certain extent. By leveraging both accelerometer and gyroscope data, we were able to deduce the displacement and orientation of the sensor. This data was then emulated to visually reconstruct movements on a cube object within Unity game engine, post which manual classification was attempted to validate the reconstructed data against ground truth labels. The validation process showed that our proposed system was comprehensible by 24 survey participants with 63% accuracy.

6 Future Work

Moving forward, an important part of future work involves extending our visual reconstruction methodology to reconstruct synthetically generated data. This will help us validate the synthetic data and aid in the generation of extensive HAR datasets with the help of generative networks.

To address the limitations identified in our study, we plan to explore advanced filtering techniques and improved sensor fusion algorithms and blend them together. Since the foundation of visual reconstruction relies on the conversion of accelerometer and gyroscope data to displacement and orientation data, minimizing errors and optimizing this conversion process is crucial to the accuracy, comprehensibility and consistency of our visual reconstructions.

Moreover, we intend to investigate a multi-sensor approach using a humanoid model in Unity. By applying forces at various joints, we can simulate and analyze how different sensor placements affect the reconstruction. While this approach provides a promising foundation for future research, it requires extensive experimentation to overcome challenges related to error accumulation and calibration issues. Rigorous testing and iterative refinement will be necessary to develop a reliable multi-sensor visual reconstruction model.

References

- [1] Z. Song, Z. Cao, Z. Li, J. Wang, and Y. Liu, “Inertial motion tracking on mobile and wearable devices: Recent advancements and challenges,” *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 692–705, 2021.
- [2] H. Duan, “haorand/awesome-human-activity-recognition,” 06 2024.
- [3] S. Shen, H. Wang, and R. Roy Choudhury, “I am a smartwatch and i can track my user’s arm,” in *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, pp. 85–96, 2016.
- [4] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.
- [5] V. Mollyn, R. Arakawa, M. Goel, C. Harrison, and K. Ahuja, “Imuposer: Full-body pose estimation using imus in phones, watches, and earbuds,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2023.
- [6] J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, and B. Eberhardt, “Motion reconstruction using sparse accelerometer data,” *ACM Transactions on Graphics (ToG)*, vol. 30, no. 3, pp. 1–12, 2011.
- [7] T. Sun, X. He, and Z. Li, “Digital twin in healthcare: Recent updates and challenges,” *Digital Health*, vol. 9, p. 20552076221149651, 2023.
- [8] H. Zhou, L. Wang, G. Pang, H. Shen, B. Wang, H. Wu, and G. Yang, “Toward human motion digital twin: A motion capture system for human-centric applications,” *IEEE Transactions on Automation Science and Engineering*, 2024.
- [9] T. Tahmid, M. A. Lobabah, M. Ahsan, R. Zarin, S. S. Anis, and F. B. Ashraf, “Character animation using reinforcement learning and imitation learning algorithms,” in *2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 1–6, IEEE, 2021.
- [10] A. Ferrari, D. Micucci, M. Mobilio, and P. Napolitano, “Trends in human activity recognition using smartphones,” *Journal of Reliable Intelligent Environments*, vol. 7, no. 3, pp. 189–213, 2021.
- [11] M. Karim, S. Khalid, A. Aleryani, J. Khan, I. Ullah, and Z. Ali, “Human action recognition systems: A review of the trends and state-of-the-art,” *IEEE Access*, vol. 12, pp. 36372–36390, 2024.
- [12] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *Pervasive Computing* (A. Ferscha and F. Mattern, eds.), (Berlin, Heidelberg), pp. 1–17, Springer Berlin Heidelberg, 2004.
- [13] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *The European Symposium on Artificial Neural Networks*, 2013.
- [14] O. D. Incel, M. Kose, and C. Ersoy, “A review and taxonomy of activity recognition on mobile phones,” *Bio-NanoScience*, vol. 3, no. 2, pp. 145–171, 2013.
- [15] M. Kok, J. D. Hol, and T. B. Schön, “Using inertial sensors for position and orientation estimation,” *arXiv preprint arXiv:1704.06053*, 2017.
- [16] S. Shen, M. Gowda, and R. Roy Choudhury, “Closing the gaps in inertial motion tracking,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pp. 429–444, 2018.
- [17] S. Madgwick *et al.*, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” *Report x-io and University of Bristol (UK)*, vol. 25, pp. 113–118, 2010.
- [18] W. R. Hamilton, “Ii. on quaternions; or on a new system of imaginaries in algebra,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, no. 163, pp. 10–13, 1844.
- [19] J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.
- [20] J. D. Hol, *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. PhD thesis, Linköping University Electronic Press, 2011.
- [21] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Padiaditis, and M. Tsiknakis, “The mobiaact dataset: Recognition of activities of daily living using smartphones,” in *International conference on information and communication technologies for ageing well and e-health*, vol. 2, pp. 143–151, SciTePress, 2016.
- [22] D. Micucci, M. Mobilio, and P. Napolitano, “Unimib shar: A dataset for human activity recognition using acceleration data from smartphones,” *Applied Sciences*, vol. 7, no. 10, p. 1101, 2017.

-
- [23] G. M. Weiss, K. Yoneda, and T. Hayajneh, "Smartphone and smartwatch-based biometrics using activities of daily living," *Ieee Access*, vol. 7, pp. 133190–133202, 2019.
 - [24] M. Karas, J. Urbanek, C. Crainiceanu, J. Harezlak, and W. Fadel, "Labeled raw accelerometry data captured during walking, stair climbing and driving (version 1.0. 0)," *PhysioNet*, 2021.
 - [25] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
 - [26] 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*, UbiComp/ISWC '20 Adjunct, (New York, NY, USA), p. 249–254, Association for Computing Machinery, 2020.
 - [27] W.-H. Chen and P.-C. Cho, "A gan-based data augmentation approach for sensor-based human activity recognition," *International Journal of Computer and Communication Engineering*, vol. 10, no. 4, pp. 75–84, 2021.
 - [28] Y. Desmarais, D. Mottet, P. Slangen, and P. Montesinos, "A review of 3d human pose estimation algorithms for markerless motion capture," *Computer Vision and Image Understanding*, vol. 212, p. 103275, 2021.
 - [29] T. D. Nguyen and M. Kresovic, "A survey of top-down approaches for human pose estimation," *arXiv preprint arXiv:2202.02656*, 2022.
 - [30] T. Van Wouwe, S. Lee, A. Falisse, S. Delp, and C. K. Liu, "Diffusionposer: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2513–2523, 2024.
 - [31] M. Al Borno, J. O'Day, V. Ibarra, J. Dunne, A. Seth, A. Habib, C. Ong, J. Hicks, S. Uhlich, and S. Delp, "Opensense: An open-source toolbox for inertial-measurement-unit-based measurement of lower extremity kinematics over long durations," *Journal of neuroengineering and rehabilitation*, vol. 19, no. 1, p. 22, 2022.
 - [32] P. Ge, J. Hong, and Y. Ping, "Research on an inertial sensor-based human motion twin system," in *NCIT 2022: Proceedings of International Conference on Networks, Communications and Information Technology*, pp. 1–10, VDE, 2022.
 - [33] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," 2020.
 - [34] D. Laidig, "dlaidig/qmt," 05 2024.
 - [35] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.
 - [36] "GitHub - xioTechnologies/Fusion — github.com." <https://github.com/xioTechnologies/Fusion?tab=readme-ov-file>. [Accessed 03-08-2024].
 - [37] M. Heydarian and T. E. Doyle, "rwisdm: Repaired wisdm, a public dataset for human activity recognition," *arXiv preprint arXiv:2305.10222*, 2023.
 - [38] A. Blake, G. Winstanley, and W. Wilkinson, "Deriving displacement from 3-axis accelerometers," in *Computer Games, Multimedia & Allied Technology*, 2009.
 - [39] "Reconstruction of human activity from synthetically generated data using inertial motion sensors — docs.google.com." <https://docs.google.com/forms/d/1zWCbY2wK9D8ePr4tbDj4igLexphQUdxXsWCb8oQGxTI/edit>. [Accessed 08-08-2024].

Appendix

Technical details

For data analysis, visualisation and data preparation, we utilize Python 3.11.9. The development environment is set up using Visual Studio Code (VS Code) 1.91.1. A key library which we utilize in our data preparation process is the *imufusion* library, version 1.2.5 (36) for implementing sensor fusion. This library is instrumental in converting raw accelerometer and gyroscope data into meaningful orientation data represented by euler angles. The *imufusion* library implements an Attitude and Heading Reference System (AHRS) algorithm which integrates gyroscope data with feedback from other sensors, functioning as a complementary filter to reduce noise and enhance the accuracy of orientation measurements. This conversion process is critical for accurately representing the motion data in subsequent visualization steps.

For the visual reconstruction of the processed motion data, we employ Unity 2022.3.29f1, a leading game engine renowned for its capabilities in creating interactive and immersive 3D environments. Unity provides a robust platform for real-time rendering and simulation, making it an ideal choice for our project. We write the scripts for visual reconstruction in C#, the inbuilt scripting language of Unity. This allows us to leverage Unity's powerful features for object manipulation, animation, and real-time physics simulation.

Github Repository

<https://github.com/koustavbarik14/HumanActivityReconstruction>

Abbreviations

The following abbreviations are used in this manuscript:

HAR - Human Activity Recognition

IMU - Inertial Measurement Unit

LRF - Local Reference Frame

GRF - Global Reference Frame

CSV - Comma Separated Values