

Mapping Vicon Motion Tracking to 6-axis IMU Data for Wearable Activity Recognition

Lloyd Pellatt, Alex Dewar, Andrew Philippides and Daniel Roggen

Abstract There are several large datasets available captured from motion tracking systems which could be useful to train wearable human activity recognition (HAR) systems, if only their spatial data could be mapped into the equivalent inertial measurement unit (IMU) data that would be sensed on body. In this paper we describe a mapping from 3D Vicon motion tracking data to data collected from a BlueSense on-body IMU. We characterise the error incurred in order to discern the extent to which it is possible to generate useful training data for a wearable activity recognition system from data collected with a motion capture system. We analyze this by mapping Vicon motion tracking data to rotational velocity and linear acceleration at the head, and compare this to actual gyroscope and accelerometer data collected by an IMU mounted on the head. In a 15 minute dataset comprising three static activities —sitting, standing and lying down—we find that 95% of the reconstructed gyroscope data is within an error of $[-7.25;+7.46] \text{ deg} \cdot s^{-1}$, while 95% of the reconstructed accelerometer data was contained within $[-96.1;+72.9] \text{ m} \cdot G$. However, when we introduce more movement by including data collected while walking this increases to $[-19.0;+18.2] \text{ deg} \cdot s^{-1}$ for the gyroscope and $[-208;+186] \text{ m} \cdot G$ for the accelerometer. We conclude that generating accurate IMU data from motion capture datasets is possible and could be useful in providing larger volumes

Lloyd Pellatt
Sensor Technology Research Centre, University of Sussex, Brighton, United Kingdom, e-mail:
lp349@sussex.ac.uk

Dr. Alex Dewar
Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, United Kingdom, e-mail: A.Dewar@sussex.ac.uk

Prof. Andrew Philippides
Centre for Computational Neuroscience and Robotics, University of Sussex, Brighton, United Kingdom, e-mail: andrewop@sussex.ac.uk

Prof. Daniel Roggen
Sensor Technology Research Centre, University of Sussex, Brighton, United Kingdom, e-mail:
daniel.roggen@ieee.org

of data for activity recognition tasks and in helping enable advanced, data-hungry techniques such as deep learning to be employed on a larger scale within the domain of human activity recognition.

1 Introduction

One challenge faced by sensor-based HAR systems is the difficulty involved in collecting large, well annotated datasets, with enough samples to implement state of the art techniques such as deep learning. While several excellent public datasets exist for on-body and smartphone sensor-based HAR [1], [2], [3], for more niche applications it is very difficult to find public data, and collecting such data independently can be prohibitively time-consuming and expensive.

We investigate whether it may be possible to alleviate this issue by borrowing data from the closely related field of vision-based HAR. Many vision-based HAR systems use a combination of depth sensors and RGB imaging [4] to recreate a ‘2.5D’ view of the scene, and while this technique has proved effective in many cases [5],[6], it suffers from issues with occlusion, especially when only one depth sensor/camera is used for imaging.

Motion capture systems with multiple cameras get around this occlusion problem, and are increasingly being employed for the purposes of HAR —for example [7] and [8] use a Vicon motion capture system [9] for position tracking and emotion recognition respectively. There are also several large, comprehensive datasets available publicly which provide annotated motion capture data for a range of activities [10], [11],[12],[13]. If this data could be tapped into for sensor-based HAR it would be extremely useful both in terms of finding more uses for existing data and enabling future studies with a broader scope.

In this work we propose a technique to analytically derive the angular velocity and linear acceleration at a given point on the body by considering the rotational and spatial coordinates of an object as reported by the Vicon tracking software, and characterise the error incurred in applying this transformation to a dataset of simple activities which we collected for this evaluation. We use a BlueSense [14] on-body IMU sensing platform as a reference.

The key contribution of this work is an experimental protocol and dataset, an analytical model to translate motion tracking data into 6-axis IMU data (i.e. 3D acceleration and 3D angular velocity) at a given location on the body, characterisation of the error incurred by employing this technique, and discussion of the limitations of the technique as well as open questions for future research.

2 Related Work

Some similar attempts have been made to bring vision-based HAR data into the IMU domain, notably [15] works with monocular RGB videos. Here the sensor data was generated from 2D poses extracted using OpenPose. A deep regression network trained on a dataset of videos and corresponding IMU data collected from subjects performing certain exercises and movements was used to reconstruct the orientation independent ‘acceleration norm’ of each sensor. The acceleration norm is defined as $\sqrt{x^2 + y^2 + z^2}$, where x, y, z are the accelerometer axes. They then used the trained network to reconstruct data from YouTube videos of individuals performing an exercise routine, and used this data to train an activity recognition classifier. They found that while the classifier trained on the reconstructed data alone failed to match the performance of the classifier when trained on ‘real’ data, the best performance was achieved when trained on a combination of ‘real’ and reconstructed data.

In [16], a MIMO system identification technique is used to obtain a mapping from one sensor modality to another, by defining a matrix of parameters which when applied to the original data (with a delay term) allow for a range of static and dynamic transformations including rotation and differentiation. They found that when generating acceleration data with this technique from data collected using a Microsoft Kinect motion tracking system, the accuracy of a gesture recognition model trained on the generated data was 4% below the baseline accuracy.

Other works which look to solve the difficulty of collecting large labelled datasets for HAR include [17], which uses a vision-based system to generate labels which are used to help train a sensor-based system concurrently, and [18], which leverages deep transfer learning to use selected samples from a source domain to augment data used to train a classifier on a separate target domain. The first is fundamentally different from our approach in that it targets systems with both vision and sensing capabilities, while the second targets transfer learning within the same sensor modality but applied to different body parts or different subjects.

There has also been some work which has looked into simulating IMU data, either from source data collected in a different modality [19], [20] or from a 3D simulation of a moving body [21]. In [20], IMU data is simulated based on motion capture data, and several activity recognition classifiers are characterised on the simulated data, showing similar performance to those trained on real data. The authors do not give a description or characterisation of the technique used to transform the motion capture data into IMU data, which we aim to provide in this article.

IMUSim [19] is a more rigorous construction of a simulator for 9 axis IMU measurements (consisting of accelerometer, gyroscope and magnetometer data) which can be generated from existing motion capture or other sampled data, and they model the effects of a range of environmental and system specific factors such as magnetic/electric field distortions and sensor noise on the IMU measurements. They use a similar method to ours to reconstruct IMU data, but do not give a detailed description of the method, and they apply an interpolating spline to the output in order to obtain a continuous signal from which to take a derivative, which could introduce some additional error. In contrast we will present our method in full and

apply numerical differentiation to obtain an accurate, discrete approximation of rate of turn and acceleration. We also give a characterisation of the method on a new dataset.

Bodysim [21] takes a different approach in that the IMU data is modelled based on a 3D model of a human body (a ‘human structure model’) which is initialised using a Microsoft Kinect. This is combined with a ‘human Dynamics Model’ and a ‘human to system properties relationship model’ which can be used to generate readings from sensors placed at a configurable position on the simulated body. This allows for a lot of freedom in choice of activity but does not have the same application to real world datasets as our method.

In summary, current work in mapping one modality to another mostly either relies upon the knowledge of both datastreams simultaneously (at least for training), or is targeted at generating new simulated datasets of IMU data rather than transforming existing motion capture datasets. Nonetheless results from other work have shown that transferring knowledge from other modalities is possible with minimal effects on classifier performance. To our knowledge the only other work describing an analytical technique to transform data from motion capture to IMU format is IMUSim [19], which however did not characterise the method in detail, preferring to focus on other factors such as gravitational and magnetic field distortions which would affect the final simulated data. We aim to provide a thorough explanation of the technique as well as a characterisation of its effectiveness on a novel dataset. An analytical approach has its own limitations as we will discuss later on in this paper, but has the advantage of potentially being very robust and generalisable, since we rely on physical relationships between different modalities rather than learned correlations, which can be vulnerable to bias in the training set.

3 Methodology

In this work we explicitly derive the coordinate system transformation and subsequent differentiation which allows us to reconstruct IMU data from data collected using a Vicon motion tracking system. We do this using an analytical method, rather than learning the mapping directly from the data with a technique such as deep learning. The advantage of our method over deep learning is that we do not need to use a labelled training set —we only need to know the orientation of our target system at one point in the data. In addition, a learned model would necessarily be dataset specific, where we aim to characterise an analytical method which is generalisable to any HAR motion capture dataset which provides information about the orientation of the subject.

We first collected data from several on-body IMUs and a Vicon system concurrently. Our proposed approach identifies the orientation of the target sensor relative to the motion capture object in order to find an appropriate rotational transform to align the coordinate systems of the two devices.

Once the coordinate systems are lined up, we derive the angular velocity and linear acceleration (accounting for the Earth's constant gravitational acceleration) along each axis of the combined coordinate system, which will give us an approximation of the sensor readings reported by the gyroscope and accelerometer of the IMU respectively.

3.1 Vicon Motion Tracking System

The Vicon system consists of a collection of 8 infra-red cameras mounted such that each has a full view of the space contained within the cameras, as seen in figure 2. It is an example of a motion tracking system which uses reflective markers to provide spatial information about a body —by tracking each marker individually in each cameras field of view with respect to an absolute origin which is seen by every camera, the system can then reconstruct the position of the marker in 3D space. The Vicon system also provides an option to define a collection of markers as an ‘object’, and can then infer both translational and rotational information about the center of gravity of the object with reference to the initial position and the Vicon global coordinate system. The output of a Vicon recording session reports the translational x, y and z coordinates of each tracked object in the session, plus rotational information - either Euler angles or quaternions. Here we use the quaternion representation as it gives a more robust physical description of a rotating system, avoiding singularities (gimbal lock). We discuss quaternions further in section 3.3. The data is collected at a rate of 100Hz.

3.2 BlueSense IMU

The BlueSense IMU sensing platform contains a 9-axis IMU, measuring the rotational velocity and linear acceleration along the axes shown in 1 as well as the magnetic field strength and direction. This data is then processed using Madgwick's algorithm [22] to give the orientation of the device with respect to the Earth's magnetic field, with magnetic north in the positive x direction, positive y being west and the positive z direction being opposite to the direction of gravity as inferred from the accelerometer data. The IMU collects data at up to 500Hz, which is the setting we used for this data collection.

3.3 Quaternions

Quaternions are an extension of the complex number system, expressed in the form $q = a + bi + cj + dk$, where a is a real number and i, j, k are imaginary symbols which

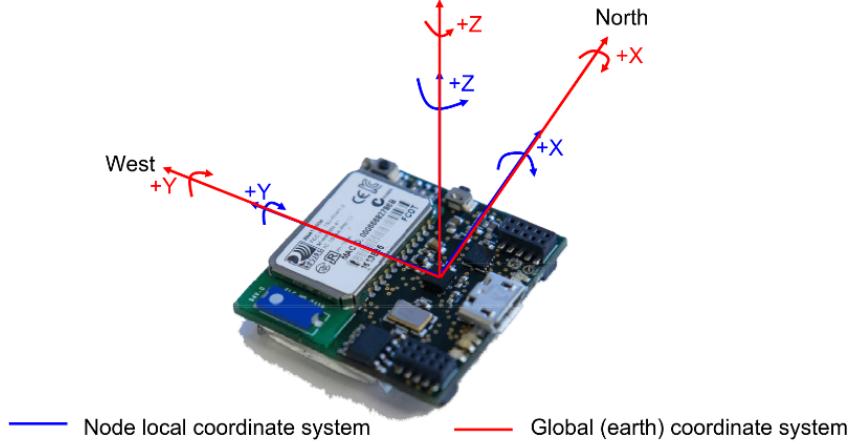


Fig. 1: The coordinate system of the BlueSense. In this figure, BlueSense is aligned to the Earth reference coordinate system, i.e. it has no rotation relative to the reference coordinate system.

are defined by the identity $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i} \cdot \mathbf{j} \cdot \mathbf{k} = -1$. They can be used to represent rotations in 3D space by realising that the real part can be interpreted as a scalar in \mathbb{R} and the imaginary part as a vector in a 3 dimensional space, corresponding to a rotation by an angle $\alpha = \frac{q}{2}$ about an axis $\mathbf{u} = (b, c, d)$. This rotation can be applied to a vector \mathbf{v} to produce the rotated vector \mathbf{v}' by taking

$$\mathbf{v}' = q\mathbf{v}q^{-1}. \quad (1)$$

The product of two quaternions gives the composition of the two rotations, and such quaternion products are non-commutative —a rotation by q and then p is the same as rotation by pq , but qp represents a rotation by p followed by q .

3.4 Experimental Setup

We collected data from 3 BlueSense devices simultaneously with the Vicon data, mounted on the head, hip and wrist of the subject, . In addition to these, we also collected data from an eSense earable device [23] as a second reference point for the head. We use the Vicon system to define objects representing the head, waist and forearm of the subject by placing markers on the body as shown in figure 2b. To synchronize the data, we set the internal clock of each Bluesense to the same value. We also performed a synchronisation gesture at the beginning of the data collection in order to synchronize the Vicon/eSense and BlueSense devices after the fact, by

manually aligning the data streams using the synchronisation gesture as a reference. Figure 2a shows the camera setup.

3.5 Experimental Protocol

We collected data from a single subject performing a sequence of 4 activities - walking, sitting, standing and lying down. These simple activities could be relevant in a home assistance scenario. In total we evaluated our method on 792 seconds of data consisting of 282 seconds standing, 280 seconds sitting, 69 seconds walking, and 161 seconds lying down. This corresponds to 396000 samples of IMU data collected at 500 Hz and 79200 samples of Vicon data collected at 100 Hz.

3.6 Data Analysis

In this section we describe the techniques used to manipulate the data. The data reported by the vicon system consists of translational X,Y and Z coordinates of the center of each object, as well as rotational information encoded by quaternions representing the shortest rotation from the initial position of the object to the position of the object at a given time. From this we attempted to generate data corresponding to the outputs of a 6-axis IMU mounted on the body at the centre of the vicon object. The reconstruction process consists of two steps - first, the data must be transformed from the coordinate system defined by the vicon system into the coordinate system of the target sensor. These can be seen in figure 2a, where the coordinate system of the BlueSense is as shown in figure 1 rotated appropriately.

Since the data is collected at two different sample rates (100Hz Vicon data, 500Hz BlueSense IMU data), in order to synchronize the two datastreams we downsample both to 50Hz, simultaneously applying a low pass filter with a cutoff frequency of 100Hz so as to retain the most information possible.

3.6.1 Transformation of Coordinate System

In order to generate meaningful approximations of the quantities which would be reported by each of the sensors, we must first align the coordinate system of the vicon data to that of the sensor. The coordinate system of the BlueSense platform is given in figure 1, while the coordinate systems of the Vicon system and the BlueSense when placed on the head are shown in figure 2b. In order to align the coordinate systems we recognise that each system has an orientation, compared to an arbitrary reference coordinate system, encoded by a quaternion \mathbf{q}_V for the Vicon and \mathbf{q}_{IMU} for the IMU. This means that, for any absolute vector \mathbf{u} in a universal coordinate system, the two systems will report



Fig. 2: (a) The Vicon setup, consisting of 8 cameras surrounding a tracking area of around 3x6 meters. (b) The coordinate system of the head mounted BlueSense IMU. Additional IMUs are mounted at the wrist and hip, and the eSense is visible in the picture, in the ear opposite the head mounted BlueSense. (c) Walking and sitting activities.

$$\mathbf{u}_{IMU} = \mathbf{q}_{IMU} \mathbf{u} \mathbf{q}_{IMU}^{-1} \quad (2)$$

and

$$\mathbf{u}_V = \mathbf{q}_V \mathbf{u} \mathbf{q}_V^{-1} \quad (3)$$

for the IMU and Vicon respectively (from equation 1). In order to reconstruct the IMU data from the Vicon data, we must then find a transformation which maps \mathbf{u}_V onto \mathbf{u}_{IMU} , i.e. some rotation \mathbf{q}_{diff} such that:

$$\mathbf{q}_{diff} \mathbf{u}_V \mathbf{q}_{diff}^{-1} = \mathbf{u}_{IMU}. \quad (4)$$

This is equivalent to:

$$\mathbf{q}_{diff} \mathbf{q}_V \mathbf{u} \mathbf{q}_V^{-1} \mathbf{q}_{diff}^{-1} = \mathbf{q}_{IMU} \mathbf{u} \mathbf{q}_{IMU}^{-1}. \quad (5)$$

Since a quaternion product is equivalent to a composition of two rotations, we can therefore say that

$$\mathbf{q}_{diff} \mathbf{q}_V = \mathbf{q}_{IMU}, \quad (6)$$

and finally

$$\mathbf{q}_{diff} = \mathbf{q}_{IMU} \mathbf{q}_V^{-1}. \quad (7)$$

The quaternion \mathbf{q}_{diff} will be constant as long as the orientation of the sensor relative to the body remains constant. Any vector quantity measured by the IMU can be generated from the equivalent quantity measured by the Vicon by applying this rotation, and the quaternions themselves can be predicted using equation 6.

\mathbf{q}_{diff} can be obtained in two distinct manners. In the first case, if the relationship between the coordinate system of the motion capture system and the global coordinate system is known (for example if the creators of the motion capture dataset publish their coordinate system with reference to cardinal directions), then we can infer \mathbf{q}_{diff} by measuring the quaternion reported by the IMU worn on the body when facing in the direction of one of the principal axes, then contrast this with a sample from the motion capture data where the subject is positioned similarly.

In the second case, where the relationship between the coordinate system of the motion capture system and the global coordinate system is unknown, it may be necessary to use a trial and error approach - if the chosen orientation is very wrong there will likely be some discontinuities in the data, where if it is close there will simply be a systematic offset in the simulated data. It is also possible that the target system will have some user-specific variability in wearing position. This would introduce another uncertainty, which would have to be taken into account when designing any classification system.

In this work, we use the first option to simulate the IMU data which would be captured at the center of mass of the Vicon object. In order to generate data from, say, a wearable positioned on the edge of the head, some additional work would be

required especially when considering the gyroscope, since the angular velocity will be larger at the periphery of the body. However, we show that this effect is small enough to not contribute too much error to the transformation.

3.6.2 Reconstruction of Gyroscope Data

In order to derive the rotational velocity of the sensor, some further computation is required. Since we only have the change in orientation of the Vicon object over time in terms of the quaternions themselves, we must project these changes onto the axes of the target sensor. Once we have reconstructed the quaternions of the IMU as described in equation 6, we can derive the rotational velocity about each axis of the sensor by considering the change in the quaternions over time.

First, we find the quaternion difference $d\mathbf{q}_t$, which represents the rotation between the quaternions at times $t - 1$ and t in the same way as in equation 7

$$d\mathbf{q}_t = \mathbf{q}_t \mathbf{q}_{t-1}^{-1}. \quad (8)$$

Then, for each axis of our coordinate system rotated at each timestep by $d\mathbf{q}_t$, we want to find the rotation experienced by the body about each of the three axes, which we can then divide by the time difference between the two steps in order to obtain our rotational velocity. This is equivalent to saying we want to find the projection of the 3D-rotation between each time step onto the plane which is perpendicular to each of the three axes (basis vectors). We can achieve this for the x axis defined by the basis vector \mathbf{x} by first rotating a perpendicular vector \mathbf{n} by $d\mathbf{q}_t$ (\mathbf{n} can be any vector in the plane defined by the other two basis vectors, i.e. whose component in the x direction is zero) to find $\mathbf{n}' = d\mathbf{q}_t \mathbf{n} d\mathbf{q}_t^{-1}$, and then subtracting the component of the rotation which is parallel to \mathbf{x} . This can be found by taking the dot-product of the rotated vector with the basis vector of the axis whose rotation we want to find, so

$$\mathbf{n}' = d\mathbf{q}_t \mathbf{n} d\mathbf{q}_t^{-1} \quad (9)$$

$$\theta_x = \mathbf{n}' - \mathbf{n}' \cdot \mathbf{x}, \quad (10)$$

where θ_x is the total rotation about the axis defined by the basis vector \mathbf{x} between times t and $t - 1$. To find the rotational velocity we then divide this by the magnitude of the time difference. Thus the rotational velocity ω_x about the x axis is

$$\omega_x = \frac{d\mathbf{q}_t \mathbf{n} d\mathbf{q}_t^{-1} - (d\mathbf{q}_t \mathbf{n} d\mathbf{q}_t^{-1}) \cdot \mathbf{x}}{T_t - T_{t-1}}, \quad (11)$$

where \mathbf{n} is again any vector in the plane defined by the other two basis vectors (\mathbf{y} and \mathbf{z}), and T_t refers to the time at which sample t was acquired. This can be repeated for each axis by choosing an appropriate basis vector and perpendicular vector, and in this way we can build up the total difference in angle $\theta_t = (\theta_x, \theta_y, \theta_z)$ and angular velocity $\omega_t = (\omega_x, \omega_y, \omega_z)$ between the samples t and $t - 1$.

3.6.3 Reconstruction of Accelerometer Data

The acceleration reported by an IMU at any time is a linear combination of the dynamic acceleration due to the movement of the sensor and the constant acceleration due to the Earth's gravity. The dynamic part \mathbf{a}_{dt} can be obtained simply by performing a numerical second derivative on the position data reported by the motion capture system and then rotating the result by \mathbf{q}_{diff} . Here \mathbf{v}_{dt} refers to the dynamic velocity at time t and \mathbf{a}_{dt} refers to the dynamic acceleration at time t :

$$\mathbf{v}_{dt} = \mathbf{q}_{diff} \frac{\mathbf{x}_{vt} - \mathbf{x}_{vt-1}}{T_t - T_{t-1}} \mathbf{q}_{diff}^{-1} \quad (12)$$

$$\mathbf{a}_{dt} = \mathbf{q}_{diff} \frac{\mathbf{v}_{dt} - \mathbf{v}_{dt-1}}{T_t - T_{t-1}} \mathbf{q}_{diff}^{-1}. \quad (13)$$

The acceleration due to gravity as seen by the sensor is more complicated - although it will always sum (in quadrature) to G, its distribution across the axes of the sensor is dependent on the orientation of the device.

We know that in the global coordinate system and in the coordinate system of the Vicon, the acceleration due to gravity is $\mathbf{g} = (0, 0, -9.81)ms^{-2}$. In order to predict the sensor reading, we rotate this by the quaternion reported by the Vicon system at each timestep, then by \mathbf{q}_{diff} , giving the acceleration due to gravity in the sensor frame

$$\mathbf{g}_{IMU_t} = \mathbf{q}_{diff} \mathbf{q}_{vt} \mathbf{g} \mathbf{q}_{vt}^{-1} \mathbf{q}_{diff}^{-1}. \quad (14)$$

Finally, we sum the dynamic acceleration and the acceleration due to gravity at each timestep to give our sensor prediction \mathbf{a}_{IMU}

$$\mathbf{a}_{IMU_t} = \mathbf{a}_{dt} + \mathbf{g}_{IMU_t}. \quad (15)$$

3.7 Filtering and Preprocessing Steps

We initially downsample each datastream to 50Hz, simultaneously applying an 8th order Chebyshev low-pass filter with a cutoff frequency of 25Hz. We then apply a Savitsky-Golay filter to the downsampled data to lessen the effect of discontinuities where the Vicon system lost the object momentarily, or adjusted the orientation suddenly. We discuss this issue in section 5. An example of this is shown in figure 3, where the raw Vicon data is plotted along with the filtered data, and 3b shows the acceleration computed on the raw data and the filtered data.

Finally, we downsample each signal again to 25Hz to perform the evaluation. Figure 4 shows the reference traces from the bluesense alongside the reconstructed angular velocity / linear acceleration from the raw data, as well as the reconstructed traces from the post-filtering data. We can see that the filtering eliminates a lot of unwanted noise such as that caused by the discontinuities, but also may remove some

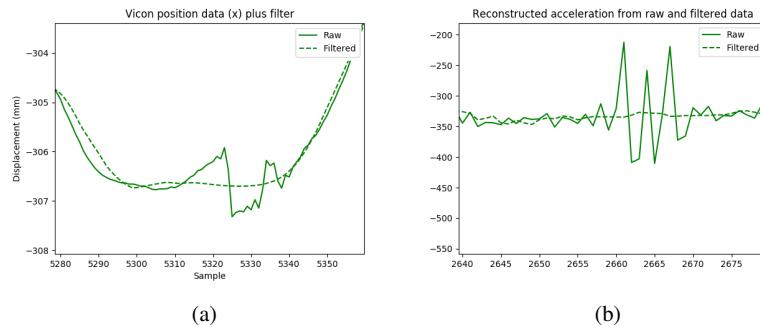


Fig. 3: (a) Vicon position data and filtered data. (b) Effect of filtering the position data with the Savitsky-Golay filter on the final reconstructed acceleration, after downsampling to 50Hz.

useful high-frequency (low-pass and Savitsky-Golay) and high-amplitude (Savitsky-Golay) information in the raw signal.

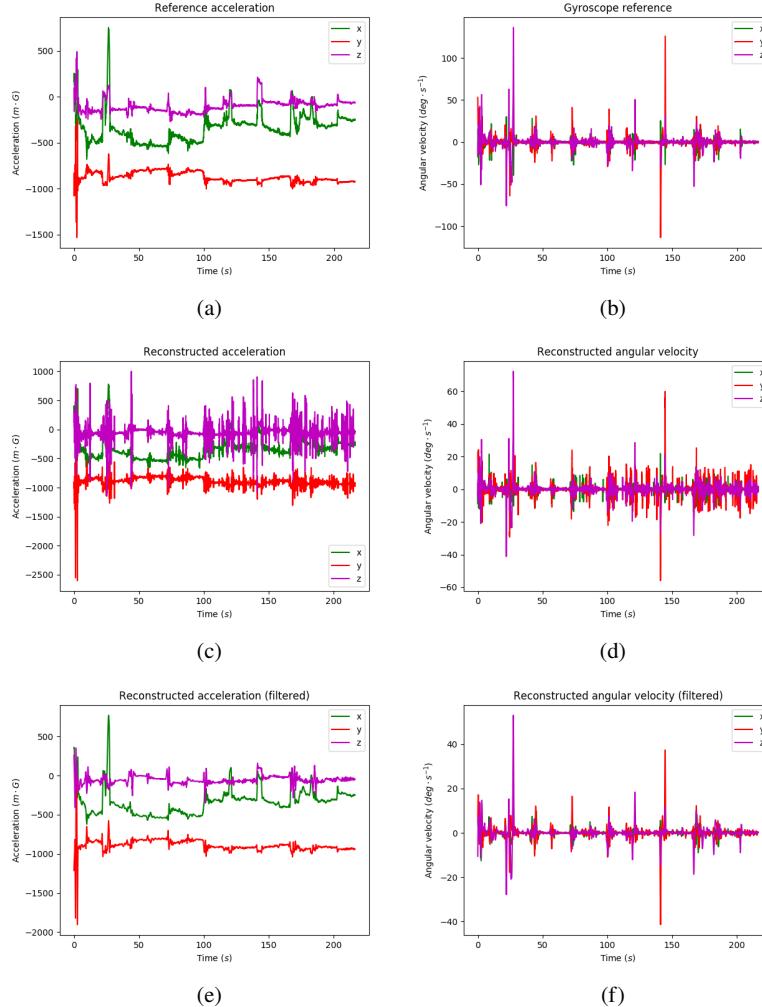


Fig. 4: The effects of applying the Savitsky-Golay filter to the Vicon data before computation of the predicted IMU data. (a) Accelerometer reference data from the head-mounted BlueSense. (b) Gyroscope reference data from the head-mounted BlueSense. (c) Accelerometer data reconstructed from the motion capture data without the Savitsky-Golay filter. (d) Gyroscope data reconstructed from the motion capture data without the Savitsky-Golay filter. (e) Accelerometer data generated using the motion capture data filtered with the Savitsky-Golay filter. (f) Gyroscope data generated using the motion capture data filtered with the Savitsky-Golay filter.

4 Results

Table 1 shows the Root Mean Square Error (RMSE) and standard deviation of the reconstructed data for each activity class and each sensor channel. We also give the upper and lower bounds within which 95% of the errors for each channel are contained. We can see that walking contributes around 3x the error to the combined results than the other activities.

We can see from the graphs in figure 4 that although the reconstructed gyroscope trace follows the same shape as the real data, the magnitude is reduced especially at fast turning events. This is likely because we generate the data from the change in orientation of the center of mass of the head, where the earphone is placed at the outer edge of the head. This source of error would also have an effect on the accelerometer readings, but to a lesser extent. It could be mitigated by modelling the spatial extent of the object in 3D and deriving the velocity at the edge.

5 Discussion

The main challenge we faced was due to sub-optimal setup of the Vicon markers—in order to correctly infer the orientation of an object, the Vicon system requires its markers be placed as asymmetrically as possible. Because the markers are not distinguishable from one another by the system, the only way the orientation can be inferred is by looking at the shape of the object in space. This relies on the shape of the markers being unique for each unique orientation, which can only be achieved if the distribution of markers is completely asymmetrical. Our marker set up was not as asymmetric as it could be, and this led to the system getting confused as to the orientation of the head, especially during the walking part of the data collection. This meant that a lot of the data was unusable (we collected 5 minutes of walking data but were only able to use 70 seconds), and is likely a large source of error in the final evaluation, since the 70 seconds we did use still contained some inconsistencies. In addition, occasionally one or more markers would be obscured from all of the cameras by a body part, which led to some sporadic discontinuities in the data and led us to choose to filter the data using a Savitsky-Golay filter as an interpolation technique, shown in figure 3.

Although our method is analytically derived and therefore theoretically applicable to any 3D motion capture system which provides orientation information about an object which can be interpreted in quaternion form, it requires knowledge of the orientation of the target sensor, which could be a limitation when converting large datasets as the data would all need to be converted for each different sensor position.

Due to the orientation measurement problem, around 8 minutes of data was not usable for the evaluation. Although we only collected data from one user, we do not believe collecting data from multiple users is key to the characterisation of this technique, since the accuracy of the reconstruction should only be affected by the energy level of the activity and not the specific way in which the activity is

Table 1: Error results for each activity plus all activities combined. For the gyroscope, units are $\text{deg} \cdot \text{s}^{-1}$, while the accelerometer units are in $\text{m} \cdot \text{G}$.

Sensor	Activity	Axis	RMSE	Standard Dev.	95% boundary
Gyro	Sitting	X	0.342	4.94	[-7.66,8.72]
		Y	0.108	5.00	[-6.85,6.08]
		Z	0.0510	6.39	[-8.91,10.9]
	Accelerometer	X	20.5	46.5	[-78.4,80.7]
		Y	4.67	52.3	[-77.0,66.4]
		Z	185	68.9	[-37.7,245]
Gyro	Standing	X	0.236	4.11	[-8.40,9.00]
		Y	0.0154	5.92	[-11.4,10.4]
		Z	0.190	6.03	[-7.73,10.7]
	Accelerometer	X	2.17	56.3	[-56.8,75.3]
		Y	12.4	89.8	[-61.5,42.2]
		Z	31.7	80.6	[-161,160]
Gyro	Lying	X	0.202	0.878	[-1.63,1.62]
		Y	0.242	1.03	[-1.22,2.63]
		Z	0.0155	1.85	[-1.24,1.15]
	Accelerometer	X	12.6	6.07	[-18.8,-3.79]
		Y	2.99	14.7	[-16.3,11.7]
		Z	0.404	6.73	[-13.8,9.76]
Gyro	Walking	X	3.75	23.9	[-38.2,71.2]
		Y	20.5	65.2	[-230,44.8]
		Z	0.575	25.8	[-53.7,39.1]
	Accelerometer	X	8.64	193	[-413,384]
		Y	13.0	207	[-360,503]
		Z	24.5	161	[-271,333]
Gyro	Combined	X	0.569	8.88	[-13.3,18.3]
		Y	1.72	22.8	[-24.3,15.1]
		Z	0.0317	10.7	[-22.2,21.1]
	Accelerometer	X	10.6	111	[-205,219]
		Y	7.42	98.7	[-196,157]
		Z	79.0	113	[-143,236]

performed. Ideally, we would collect more data from a wider range of activities including more dynamic ones, in order to better characterise the error in high-movement scenarios. Although we have presented an error between the reconstructed data and the reference, it should be noted that the BlueSense device has never been characterised in terms of its accuracy, and we do not have empirical data on how well the Vicon system is calibrated. This means that it is difficult to make any predictions about whether the error comes from the source system, the reference system or the method itself. However, the evaluation we have presented shows that our technique could be applicable to certain home activity datasets such as the TUM Kitchen dataset [13], which could now be used to train wearable activity recognition systems thanks to our work.

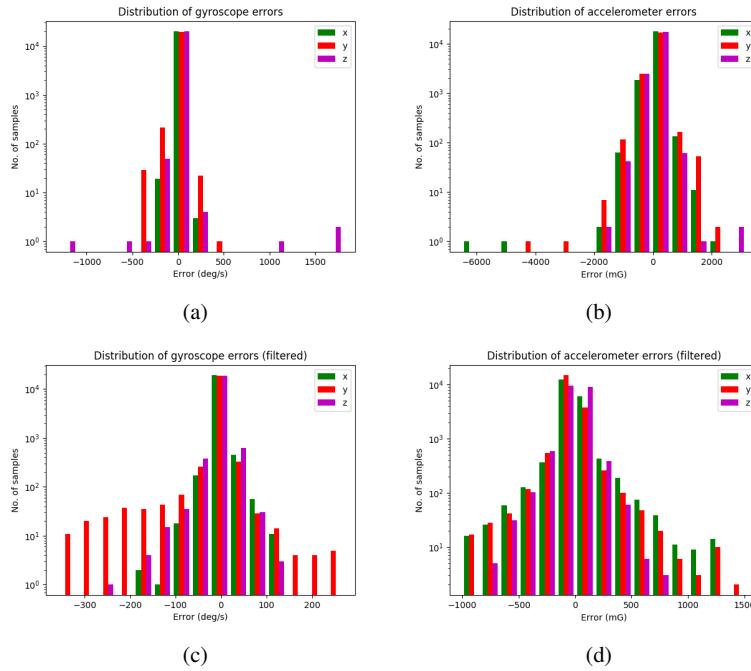


Fig. 5: (a) Distribution of errors on the reconstructed gyroscope data compared to the reference. (b) Error distribution on the reconstructed accelerometer data compared to the reference. (c) Distribution of errors on the reconstructed gyroscope data when a Savitzky-Golay filter is applied to the raw Vicon data before processing, compared to the reference. (d) Distribution of errors on the reconstructed accelerometer data when a Savitzky-Golay filter is applied to the raw Vicon data before processing, compared to the reference.

Although we have reported an error of up to $200 \text{ } m \cdot G$ in some samples, it is also interesting to note that some works have been produced which successfully trained HAR systems on quantised acceleration data [24] [25], which depending on the bin size could introduce a comparable error. We have developed and fully characterised an analytical method which can generate 6-axis IMU data from 3D motion capture datasets, which has not been done before. Similar efforts in related works include [16] and [15], which generate acceleration data from motion capture data using a learned model and IMU data from 2D RGB videos via pose estimation, respectively. In addition, [19] and [21] generate IMU data from simulated 3D bodies, generated either from motion capture or from a ‘Human dynamics model’. We cannot directly compare our error to these methods, since they do not characterise the error in absolute terms, only in terms of the effect on the accuracy of a classifier. We suggest that a possible future extension of this work would be to similarly characterise the performance of a classifier trained on IMU data reconstructed using our technique.

6 Conclusions

We have proposed an analytical method for mapping 3D motion tracking data to 3D acceleration and angular velocity as measured by an IMU at a given position and orientation relative to the mocap data. We have tested and characterised the technique using a dataset collected for this work, and found that 95% of our reconstructed IMU data falls within an error of $[-7.25;+7.46] \text{ deg} \cdot s^{-1}$ compared to our reference gyroscope and within an error of $[-96.1;+72.9] \text{ m} \cdot G$ compared to our reference accelerometer. We also found that when more movement is incorporated into the testing set, the 95% error bounds increase drastically to $[-19.0;+18.2] \text{ deg} \cdot s^{-1}$ for the gyroscope and $[-208;+186] \text{ m} \cdot G$ for the accelerometer.

Our method has the advantage of being analytically derived and therefore applicable to any scenario where IMU data must be generated from 3D motion capture data, providing that the mocap data provides orientation information which can be interpreted in quaternion form. However, this work was limited by the length and variety of our evaluation dataset, and sub-optimal setup of the Vicon markers causing sporadic discontinuities in the motion capture data. We have provided some suggestions for improvement for future work.

Our method will enable rich motion capture datasets to be converted to IMU data which can be used to train future wearable activity recognition systems. These include the CMU motion capture database [10] containing over 2600 recordings in categories such as human interactions, locomotion and sports, the MPI motion capture dataset [11] containing recordings of over 70 classes of activity, and the CMU & TUM Kitchen datasets [12] [13], a comprehensive collection of activity sequences recorded in a kitchen environment.

Acknowledgements

This work was partially funded by the EPSRC (Brains on Board project, grant number EP/P006094/1). We also thank Nvidia for their TITAN Xp donation.

References

1. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Banach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millán, “Collecting complex activity datasets in highly rich networked sensor environments,” in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, June 2010, pp. 233–240.
2. D. Micucci, M. Mobilio, and P. Napoletano, “Unimib SHAR: a new dataset for human activity recognition using acceleration data from smartphones,” *Computing Research Repository*, 2016.
3. A. Reiss and D. Stricker, “Introducing a new benchmarked dataset for activity monitoring,” in *2012 16th International Symposium on Wearable Computers*, June 2012, pp. 108–109.
4. J. Aggarwal and L. Xia, “Human activity recognition from 3D data: A review,” *Pattern Recognition Letters*, vol. 48, pp. 70 – 80, 2014.
5. A. Jalal, S. Kamal, and D. Kim, “A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments,” *Sensors*, vol. 14, no. 7, pp. 11 735–11 759, 2014.
6. S. Park, J. Park, M. Al-masni, M. Al-antari, M. Uddin, and T.-S. Kim, “A depth camera-based human activity recognition via deep learning recurrent neural network for health and social care services,” *Procedia Computer Science*, vol. 100, pp. 78 – 84, 2016.
7. C. Zhu and W. Sheng, “Realtime recognition of complex human daily activities using human motion and location data,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 9, pp. 2422–2430, 2012.
8. A. Kapur, A. Kapur, N. Virji-Babul, G. Tzanetakis, and P. F. Driessens, “Gesture-based affective computing on motion capture data,” in *International conference on affective computing and intelligent interaction*. Springer, 2005, pp. 1–7.
9. “Vicon motion tracking system documentation,” <https://docs.vicon.com/>, accessed: 2020-05-23.
10. “CMU motion capture database,” <http://mocap.cs.cmu.edu>, accessed: 2020-02-24.
11. “MPI HDM05 motion capture database,” <http://www.mpi-inf.mpg.de/resources/HDM05>, accessed: 2020-02-24.
12. “CMU kitchen data set,” <http://kitchen.cs.cmu.edu>, accessed: 2020-02-24.
13. “TUM kitchen data set,” <https://ias.in.tum.de/software/kitchen-activity-data>, accessed: 2020-02-24.
14. D. Roggen, A. Pouryazdan, and M. Ciliberto, “BlueSense - designing an extensible platform for wearable motion sensing, sensor research and IoT applications,” in *Proc. International Conference on Embedded Wireless Systems and Networks*. ACM, 2018, pp. 177–178.
15. 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*. New York, NY, USA: Association for Computing Machinery, 2019, p. 699–708.
16. O. Banos, A. Calatroni, M. Damas, H. Pomares, I. Rojas, H. Sagha, J. del R. Millán, G. Troster, R. Chavarriaga, and D. Roggen, “Kinect=IMU? learning mimo signal mappings to automatically translate activity recognition systems across sensor modalities,” in *16th International Symposium on Wearable Computers*, June 2012, pp. 92–99.
17. V. Radu and M. Henne, “Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–21, 2019.
18. J. Wang, V. W. Zheng, Y. Chen, and M. Huang, “Deep transfer learning for cross-domain activity recognition,” in *proceedings of the 3rd International Conference on Crowd Science and Engineering*, 2018, pp. 1–8.
19. 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*, 2011, pp. 199–210.

20. 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*, ser. UbiComp ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1444–1448.
21. P. Asare, R. Dickerson, X. Wu, J. Lach, and J. Stankovic, “Bodysim: A multi-domain modeling and simulation framework for body sensor networks research and design,” 09 2013, pp. 177–180.
22. S. Madgwick, A. Harrison, and R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm,” in *IEEE Int Conf on Rehabilitation Robotics*, 2011.
23. F. Kawsar, C. Min, A. Mathur, and A. Montanari, “Earables for personal-scale behavior analytics,” *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 83–89, Jul 2018.
24. M. Ciliberto, L. P. Cuspinera, and D. Roggen, “Wlcsslearn: learning algorithm for template matching-based gesture recognition systems,” in *International Conference on Activity and Behavior Computing*, S. Inoue and A. R. Ahad, Eds., vol. 1. Institute of Electrical and Electronics Engineers, February 2019, pp. 91–96.
25. L. Nguyen-Dinh, D. Roggen, A. Calatroni, and G. Tröster, “Improving online gesture recognition with template matching methods in accelerometer data,” in *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, Nov 2012, pp. 831–836.