Using Wearable Inertial Sensors for Posture and Position Tracking in Unconstrained Environments through Learned Translation Manifolds

Aris Valtazanos
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB
a.valtazanos@sms.ed.ac.uk

D. K. Arvind
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB
dka@inf.ed.ac.uk

S. Ramamoorthy
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB
s.ramamoorthy@ed.ac.uk

ABSTRACT

Despite recent advances in 3-D motion capture, the problem of simultaneously tracking human posture and position in an unconstrained environment remains open. Optical systems provide both types of information, but are confined to a restricted area of capture. Inertial sensing alleviates this restriction, but at the expense of capturing only relative (postural) and not absolute (positional) information. In this paper, we propose an algorithm combining the relative merits of these systems to track both position and posture in challenging environments. Offline, we combine an optical (Kinect) and an inertial sensing (Orient-4) platform to learn a mapping from *posture variations* to *translations*, which we encode as a *translation manifold*. Online, the optical source is removed, and the learned mapping is used to infer positions using the postures computed by the inertial sensors. We first evaluate our approach in simulation, on motion sequences with ground-truth positions for error estimation. Then, the method is deployed on physical sensing platforms to track human subjects. The proposed algorithm is shown to yield a lower average cumulative error than comparable position tracking methods, such as double integration of accelerometer data, on both simulated and real sensory data, and in a variety of motions and capture settings.

Categories and Subject Descriptors

H.4.0 [**Information Systems Applications**]: General

Keywords

Wearable inertial sensors, optical motion capture, translation models, manifold learning

1. INTRODUCTION

Many information processing applications deal with the analysis of human motion, as captured by an ensemble of sensor devices. In this context, it is often essential to determine both the absolute *position* of tracked subjects, as well as finer-grained information on their body *posture*, such as arm movements or gait patterns. Furthermore, it is often required that motion be captured in challenging, *unconstrained* areas, for example, a large office with multiple rooms and corridors, or an open outdoor environment.

The problem of simultaneous posture and position tracking is of interest to a wide range of motion capture applications. One notable domain is tracking in construction and fire-fighting missions, which requires monitoring of the deployed responders in challenging and potentially unknown environments. Position tracking systems that have been considered in this domain (e.g. indoor global positioning sensor (GPS) systems, ultra-wide bands [12], radio-frequency identification tags [11]) typically require an infrastructure that requires detailed knowledge of the environment (e.g. placing signal receivers at known positions), while also suffering from line-of-sight constraints. Addressing these needs is a challenging task when the environment is difficult to negotiate or cannot be accessed by the system designers. Furthermore, these systems cannot monitor posture, which is often important in order to determine and assess the actions performed by the deployed subjects. Similar needs arise in application areas such as physical gaming [18] and human-robot interaction in rescue missions [6], where human position and posture must also be captured in unconstrained settings. Existing techniques in these domains are similarly restricted to either providing only part of the required data, or requiring special infrastructure in order to be deployed. The above issues raise the need for tracking systems that can produce position and posture from a *single* set of sensory devices, which are not sensitive to the morphology of the capture environment, and which do not rely on synchronisation between multiple heterogeneous sources.

Despite recent advances in motion capture and sensing technologies, fulfilling the above requirements in a robust manner is a challenging task. For instance, optical motion capture systems can capture both positional and postural data, but only within contained environments limited to a small area of capture. Inertial sensing systems allow for greater flexibility in the capture environment, as they are not restricted by line-of-sight constraints between the sensing devices and the tracked subject. However, they do so at the expense of not yielding absolute positions, as their calculations are based on relative rotational estimates, e.g. gyro-

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.

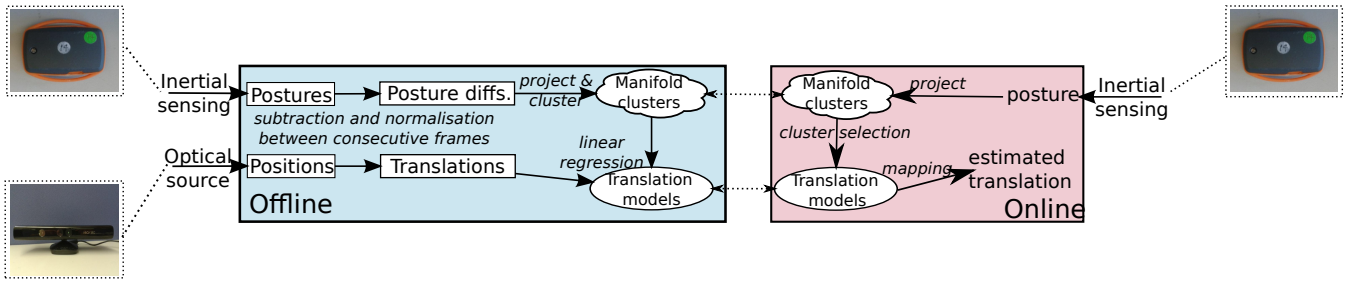


Figure 1: Overall structure of the proposed system. An inertial sensing and an optical source are synchronised and jointly used to learn generative models of whole-body translations in an offline phase. These translations are encoded as linear regression-based mappings from projected latent representations of *posture differences*, as detected from the inertial source, and *positional variations*, as detected from the optical source. Online, the optical source is removed, and the learned model is used to predict translations for the tracked subject.

scope readings. Similarly, general-purpose GPS sensors can compute absolute positions, but at a coarse level of precision and without supplying postural information, while also having limited applicability in indoor environments. Other motion capture technologies, e.g. magnetic systems, also suffer from one or multiple of the above limitations.

In order to address the challenges of simultaneous posture and position capture in unconstrained environments, one would need to combine the relative strengths of these heterogeneous systems in a principled manner. In this paper, we propose a hybrid position and posture tracking algorithm (Figure 1), which jointly uses an inertial and an optical motion capture system to learn local *models of translation* for a given human subject. The algorithm consists of an *offline* learning and *online* generation phase. In the offline phase, body posture data collected from the inertial sensors are synchronised with position data captured from the optical source and aggregated into a single dataset. Due to their high dimensionality, the posture data are projected and clustered on a low-dimensional *manifold*, which captures the salient kinematic structure of the dataset. For each cluster, the projected data are used to learn a mapping from local *posture differences* to *whole-body translations* through linear regression. In the online phase of the algorithm, the optical source is removed, and the learned models are used to generate translation vectors from estimated posture differences. By iteratively applying these translations, the proposed system can track both the position and the posture of a subject performing motions similar to those captured in the offline phase, thus overcoming the main limitation of inertial systems discussed above. Moreover, due to the removal of the optical source in the online phase, the system is not affected by the morphology of the capture environment.

In the remainder of this paper, we first review related motion capture technologies, explaining how our approach combines their relative strengths (Section 2). Then, we describe our method for posture and position tracking, distinguishing between the translation learning and generation phases (Section 3). In Section 4, our approach is evaluated in simulations and physical experiments; first, on data from the Carnegie Mellon Motion Capture Database [3], which are annotated with ground-truth positions, and then, on a human motion capture environment, where we use the Kinect and the Orient platform as sensing systems (Figure 2). Our algorithm is shown to yield a lower overall position error than the related established tracking method of acceleration integration. Furthermore, in the physical experiments, we demonstrate examples of successful position tracking in a challenging office environment, where existing motion capture technologies cannot be applied in isolation. We review the key features of our work in Section 5.

(a) Microsoft Kinect.

(b) Orient-4 device.

Figure 2: Motion tracking platforms. (a): Optical source – Kinect device with two cameras and a depth-finding sensor. (b): Inertial measurement unit – Orient-4 device with tri-axial gyroscopes, accelerometers, and magnetometers.

2. BACKGROUND AND RELATED WORK

Traditional *optical* motion capture systems, e.g. [4], use an ensemble of high-resolution cameras to track the locations of a set of reflective markers placed on the body of a subject. The marker positions are used to compute the full pose (position and posture) of the subject. However, optical systems suffer from a number of drawbacks that impact their applicability. First, motion capture must be carried out in dedicated studios, which are often expensive to set up and maintain. Second, the total area of capture is limited to a small volume, and subjects cannot be tracked outside its boundaries. Thus, optical systems cannot be used to track subjects interacting outside the studio, e.g. moving in and out of rooms, or navigating along corridors in a building. Third, occlusion problems impact the ability of these systems to track subjects consistently and reliably.

A recent development has been the creation of devices combining stereo cameras and depth-estimating sensors, notably the Kinect [1] (Figure 2(a)). The output of these sensors is used to generate a three-dimensional point cloud,

which can then be analysed to determine the pose of a tracked subject, or fit the data to a skeleton model [16]. These stereo-camera devices are significantly cheaper than traditional optical systems, and remove the need for markers on the subject's body. Furthermore, the portability of the sensors allows for anyplace motion capture. However, like traditional optical systems, these devices must remain fixed during tracking, and the volume of capture is limited to approximately 15m^3. This makes them unsuitable for tracking subjects in large or unconstrained spaces.

An alternative to optical systems are wireless *inertial* sensing platforms, such as the Orient interface [21] (Figure 2(b)). Inertial sensing systems collect data from an ensemble of sensor nodes placed on the subject's body. Each device typically consists of 3-axis inertial sensors such as gyroscopes, accelerometers, and magnetometers. These sensors jointly estimate the rotation of the body part the device is placed on, relative to a fixed point on the subject's body. Data from the different body parts are transmitted wirelessly to a base station and aggregated to determine the overall posture of the subject. Alternatively, data can also be stored on the devices and analysed offline at a later time. The latter feature makes these devices suitable for motion capture and processing in environments impacted by communication and data transmission constraints.

Due to the wireless transmission and capture of data, inertial sensing avoids the occlusion problems arising in optical systems. More importantly, as no fixed tracking source is required, subjects can be tracked in a greater variety of environments and in larger areas than with optical systems. The main drawback of inertial sensing is the relative rotational nature of estimates, which means that only postures can be determined directly. Unfortunately, this approach does not extend to absolute spatial positions, as computations are performed relative to a stationary reference point.

Position tracking using inertial measurement units has been the subject of several studies, most following a *model-based* approach, where measurements are filtered through a position model to predict the most likely translation of the tracked subject. Employed models range from Kalman [20, 8, 10] and particle filters [13], to alternative heuristic approaches based on gait event detection [15, 22, 9].

Our approach is different in being a ***model-free*** method with respect to the measurement units and their output data, where no assumptions are made on the placement of the sensors or the nature of the motion being performed. Instead, the aggregated sensory data is treated as a single *feature vector*, from which a mapping to whole-body translations is learned. This leads to an unsupervised method that can learn a model of translation without the incorporation of additional knowledge on the problem. However, the lack of a specific model also means that the quality of the learned mappings inevitably depends on the quality and variability of the training examples. In later sections of this paper, we provide illustrations of how the accuracy of our approach is affected by the provided data.

Motivated by the above features, in our experimental evaluation we compare against the established model-free method of double integration of accelerometer data. This method relies on the integration of data coming from a single sensor in order to track position over time, without having an internal model on the placement of that sensor. In Section 4, we show that using multiple sensors in conjunction with a learning algorithm can lead to better tracking of whole-body positions, in both simulated and physical experiments. Furthermore, our approach can also be combined with existing models, where the generated translations can be treated as predictive estimates for a filter. Integration with model-based filtering methods is an area of future work for us.

Dimensionality reduction has been used in inertial sensor networks as a *discriminative* model for activity recognition and gait phase detection [19, 17]. In our work, we extend this notion by using low-dimensional subspaces as *generative* models for translations. In this generative context, learned manifolds have been used in robotics, in order to facilitate imitation of human gaits by humanoid robots [14, 7]. By adopting this flexible representation, our objective is to similarly approximate a wide variety of motion dynamics.

3. METHOD

3.1 Sensory device outputs

3.1.1 Kinect

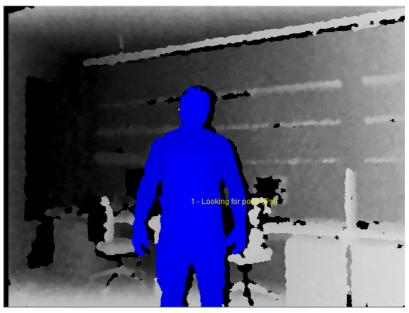


Figure 3: Body contour tracking using the Kinect. The tracking software automatically detects the outline of a human body, and tracks it as a cloud of points (shown as a blue blob).

We use the OpenNI body tracking interface [2] to detect and track the position of human subjects (Figure 3). The software automatically detects the outline of a human body, and tracks it as a collection of N_I image point coordinates, $\vec{B} = \{(x_1, y_1), \ldots, (x_{N_I}, y_{N_I})\}$. The absolute position of the tracked body is approximated as the centroid of these points as computed through *image moments*.

Let W, H be the width and height (in pixels) of the camera image, and let $\mathbf{I}$ be a 2-D array, such that

$$\mathbf{I}(a,b) = \begin{cases} 1, & (x_a, y_b) \in \vec{B} \\ 0, & (x_a, y_b) \notin \vec{B} \end{cases}, \tag{1}$$

where $1 \leq a \leq W$, $1 \leq b \leq H$. The raw image moments, M_{ij} are defined as

$$M_{ij} = \sum_{a=1}^{W} \sum_{b=1}^{H} x_a^i \cdot y_b^j \cdot \mathbf{I}(a,b). \tag{2}$$

Based on these definitions, the image coordinates of the centroid of the tracked body, $C \doteq (\bar{x}, \bar{y})$ are given by

$$(\bar{x}, \bar{y}) = (\, \texttt{round}(M_{10}/M_{00}), \texttt{round}(M_{01}/M_{00}) \,). \tag{3}$$

The depth of each image pixel (with respect to the device) is measured by the range-finding sensor of the Kinect. This information is used to convert the computed image centroid, $(\bar{x}, \bar{y})$, to the centroid of the body surface that is visible to the Kinect. These coordinates approximate to the absolute positional coordinates of the tracked body,

$$p = (x^B, y^B, z^B). \tag{4}$$

3.1.2 Orient inertial measurement units

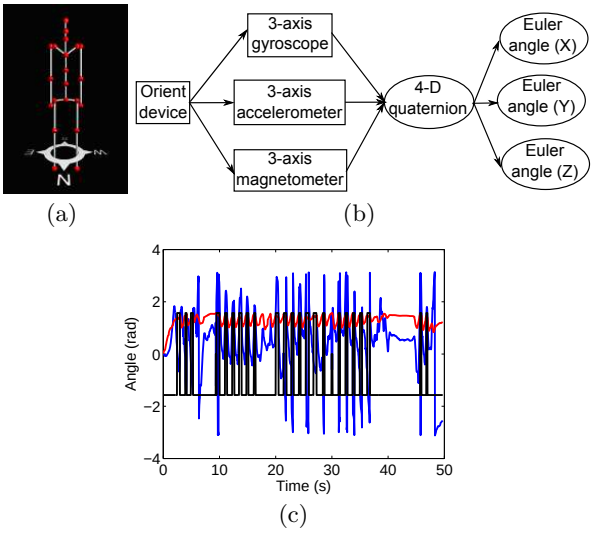


Figure 4: Posture estimation using the Orient devices. (a): 3-D model of the tracked body. Each device is mapped to a different limb. (b): Orientation estimation. Quaternions are computed from raw sensor data and converted to Euler angles. (c): Example of angles produced by an Orient device.

The posture of the tracked subject is computed by the Orient devices. Each device is placed on a limb of the subject's body (Figure 4(a)). The raw data from the device's sensors (triaxial gyroscope, accelerometer, and magnetometer) computes a quaternion representing the orientation at that limb, which is in turn converted into three-dimensional Euler angles (Figure 4(b)) based on a pre-specified rotation order. Due to this convention, an Euler angle can represent the orientation more succinctly than the corresponding quaternion, thus reducing the size of our feature set. An example of the angles output by an Orient is given in Figure 4(c). By aggregating the angles computed by all devices placed on the subject's body, we obtain the ***posture vector***

$$\pi = \{(\theta_1^x, \theta_1^y, \theta_1^z), \ \dots \ , (\theta_{N_D}^x, \theta_{N_D}^y, \theta_{N_D}^z)\}, \tag{5}$$

where N_D is the number of deployed units, and $(\theta_i^x, \theta_i^y, \theta_i^z)$ are the angles computed by the i-th unit.

3.2 Learning translation manifolds

3.2.1 Offline learning phase

In the offline phase, Kinect positions are synchronised with data from the Orient devices. From this data, a mapping from ***posture variations***, as computed by the Orient devices, to ***translations***, as computed by the Kinect, is learned through local linear regression.

Let $\{(p_1, \pi_1, t_1), \dots, (p_{\tau+1}, \pi_{\tau+1}, t_{\tau+1})\}$ be a set of recorded synchronised training data, comprising $(\tau+1)$ absolute position and posture pairs, along with the times t at which each pair was recorded. By taking the difference of successive instances, we obtain a training data set of τ *unnormalised* translations (i.e. position differences), posture variations[1], and time differences,

$$\widetilde{\mathbf{D}} = \{(\widetilde{dp}_1, \widetilde{d\pi}_1, dt_1), \ \dots \ , (\widetilde{dp}_\tau, \widetilde{d\pi}_\tau, dt_\tau)\} = \{(p_2 - p_1, \pi_2 - \pi_1, t_2 - t_1), \ \dots \ , (p_{\tau+1} - p_\tau, \pi_{\tau+1} - \pi_\tau, t_{\tau+1} - t_\tau)\}. \tag{6}$$

At this stage, translations $\widetilde{dp} = (\widetilde{dx}, \widetilde{dy}, \widetilde{dz})$ do not account for the absolute orientation of the subject's body. To address this problem, we assume that at least one inertial measurement unit, $\bar{u}$, is placed on a point where it can measure the subject's absolute orientation, $\bar{\theta}$, with respect to the transverse plane of motion. We focus on this single angle (instead of computing three-dimensional absolute orientations) because it is closely correlated with most turning movements that occur during walking motion sequences. Thus, by normalising with respect to $\bar{\theta}$, we can compensate for turns and changes of direction in the motion of the subject. We take $\bar{u}$ to be the device placed on the subject's waist or hips as a represantative location for this purpose. The required angle $\bar{\theta}$ is computed through $\bar{u}$'s magnetometers, which measure absolute orientations using Earth's magnetic field. Thus, the unnormalised translation components on the transverse plane, $(\widetilde{dx}, \widetilde{dy})$, can be normalised through the rotation

$$\begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \end{pmatrix} = \begin{pmatrix} \cos(-\bar{\theta}) & -\sin(-\bar{\theta}) \\ \sin(-\bar{\theta}) & \cos(-\bar{\theta}) \end{pmatrix} \cdot \begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \end{pmatrix}. \tag{7}$$

We also normalise translations and posture differences with respect to their recorded time intervals. Thus, the *normalised* training data set is given by

$$\mathbf{D} = \{(dp_1, d\pi_1), \ \dots \ , (dp_\tau, d\pi_\tau)\} = \{(\widetilde{dp}_1/dt_1, \widetilde{d\pi}_1/dt_1), \ \dots \ , (\widetilde{dp}_\tau/dt_\tau, \widetilde{d\pi}_\tau/dt_\tau)\}. \tag{8}$$

-Dimensionality reduction: The size of each posture variation vector, $d\pi$, is $D = 3 \cdot N_D$, where N_D is the number of deployed devices. Even if N_D is not particularly large, it may be difficult to learn a direct mapping to associated translations, due to the different modalities of the posture data. To overcome this problem, we project posture variations to a *latent space*, from which a mapping can be learned more efficiently. We use Principal Component Analysis (PCA), which embeds data into a low-dimensional *linear manifold* by maximising their variance [5]. Thus, this method seeks to preserve the high-dimensional structure of the data in the projected space. We review the key features of PCA below.

Let $\{d\pi_i\}$, $1 \leq i \leq \tau$ be the set of posture variation vectors, each having dimensionality D. The mean, $\overline{d\pi}$, and covariance matrix, $\mathbf{S}$, of these vectors are given by

$$\overline{d\pi} = \frac{1}{\tau} \sum_{i=1}^{\tau} d\pi_i, \ \ \mathbf{S} = \frac{1}{\tau} \sum_{i=1}^{\tau} (d\pi_i - \overline{d\pi})(d\pi_i - \overline{d\pi})^T, \tag{9}$$

[1]Angle differences are constrained to lie in $[-\pi, +pi)$.

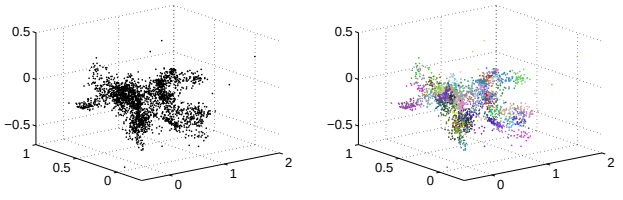
respectively. Now let d be the target dimensionality of the low-dimensional latent space, where $d < D$. We obtain the d *eigenvectors* (or principal components) of $\mathbf{S}$, $u_1, \ldots, u_d$, each of dimensionality D, corresponding to the d largest eigenvalues, $\lambda_1, \ldots, \lambda_d$ of this matrix. These vectors are set as the columns of a $D \times d$ matrix

$$\mathbf{M} = \begin{pmatrix} u_{1,1} & \cdots & u_{d,1} \\ \vdots & \ddots & \vdots \\ u_{1,D} & \cdots & u_{d,D} \end{pmatrix} \tag{10}$$

The latent representation of a D-dimensional posture variation vector $d\pi$ is given by

$$\phi = d\pi \cdot \mathbf{M}. \tag{11}$$

We refer to the manifold projections ϕ as the *feature vectors* of our translation learning algorithm. In both simulated and physical experiments (Section 4), we set the target subspace dimensionality to $d = 3$.



(a) (b)

Figure 5: Feature vector clustering example. (a): Projected points. (b): Division in 100 clusters, each represented by a different colour.

-*Feature vector clustering:* In a given set of training examples, there may be groups of similar posture variations leading to related translation vectors. To exploit this similarity, we group the projected feature vectors into clusters of related data points, and learn a separate translation mapping for each cluster (instead of a single mapping for the whole dataset). We use the k-means clustering algorithm, which groups input points into a specified number of k distinct clusters [5]. As we use clustering in a learning context, we use small (with respect to the size of the dataset) values of k to avoid overfitting the training data; in our experiments, this value does not exceed 2% of the overall number of training points. Despite the need for this manually specified parameter, k-means clustering has the advantage that it does not make assumptions about cluster structure (whereas distribution methods such as expectation-maximisation [5] assume a Gaussian form), while also favouring clusters of approximately equal sizes.

Figure 5 illustrates an example of clustering on a set of three-dimensional points. When applied on a dataset of τ feature vectors, the algorithm returns the set of clusters $\mathbf{C} = \{c_1, \ldots, c_k\}$, with centres $\vec{\mu} = \{c_1.\mu_1, \ldots, c_k.\mu_k\}$, where each cluster c_i, $1 \leq i \leq k$, consists of a set of N_i feature vectors

$$c_i = \{\phi_1^i, \ldots, \phi_{N_i}^i\}. \tag{12}$$

-*Translation mapping learning:* For each data cluster $c_i = \{\phi_1^i, \ldots, \phi_{N_i}^i\}$, we learn a mapping from its constituent feature vectors to the corresponding translations, $T^c = \{dp_1^i, \ldots, dp_{N_i}^i\}$. We learn a separate mapping for each direction of motion (x, y, z) through *linear regression* on the training points. In other words, we represent each translation component as a *linear function* of the projected feature vectors.

To learn these mappings, we collect all feature vectors of a cluster as a $N_i \times d$ design matrix,

$$\mathbf{X} = \begin{pmatrix} \phi_{1,1}^i & \cdots & \phi_{1,d}^i \\ \vdots & \ddots & \vdots \\ \phi_{N_i,1}^i & \cdots & \phi_{N_i,d}^i \end{pmatrix}. \tag{13}$$

Furthermore, we define three *observation vectors*, one for each of the directions of motion, such that

$$\mathbf{x} = \begin{pmatrix} dx_1^i \\ \vdots \\ dx_{N_i}^i \end{pmatrix}, \mathbf{y} = \begin{pmatrix} dy_1^i \\ \vdots \\ dy_{N_i}^i \end{pmatrix}, \mathbf{z} = \begin{pmatrix} dz_1^i \\ \vdots \\ dz_{N_i}^i \end{pmatrix}. \tag{14}$$

For each observation vector $\mathbf{v}$, we learn a linear mapping from the design matrix $\mathbf{X}$ using least squares approximation, represented by a set of d weights $\mathbf{w}$:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{v}\,. \tag{15}$$

By applying this procedure to all three observation vectors, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, we obtain the linear mapping weights for cluster c_i, $\mathbf{w}_x^i, \mathbf{w}_y^i, \mathbf{w}_z^i$, respectively. These can be collectively represented as the *cluster translation mapping*

$$\mathbf{W}^i = \begin{pmatrix} \mathbf{w}_{x,1}^i & \cdots & \mathbf{w}_{x,d}^i \\ \mathbf{w}_{y,1}^i & \ddots & \mathbf{w}_{y,d}^i \\ \mathbf{w}_{z,1}^i & \cdots & \mathbf{w}_{z,d}^i \end{pmatrix}, \tag{16}$$

with a different $\mathbf{W}^i$ computed for each cluster. Thus, the latent space becomes a ***translation manifold*** that can generate translations from given feature vectors.

3.2.2 Online translation generation

Learned mappings can be applied to novel instances of posture variations and predict whole-body translations. Assuming a known initial estimate of position, (x_0, y_0, z_0) and orientation, θ_0, predicted translations can be chained together to track position over time.

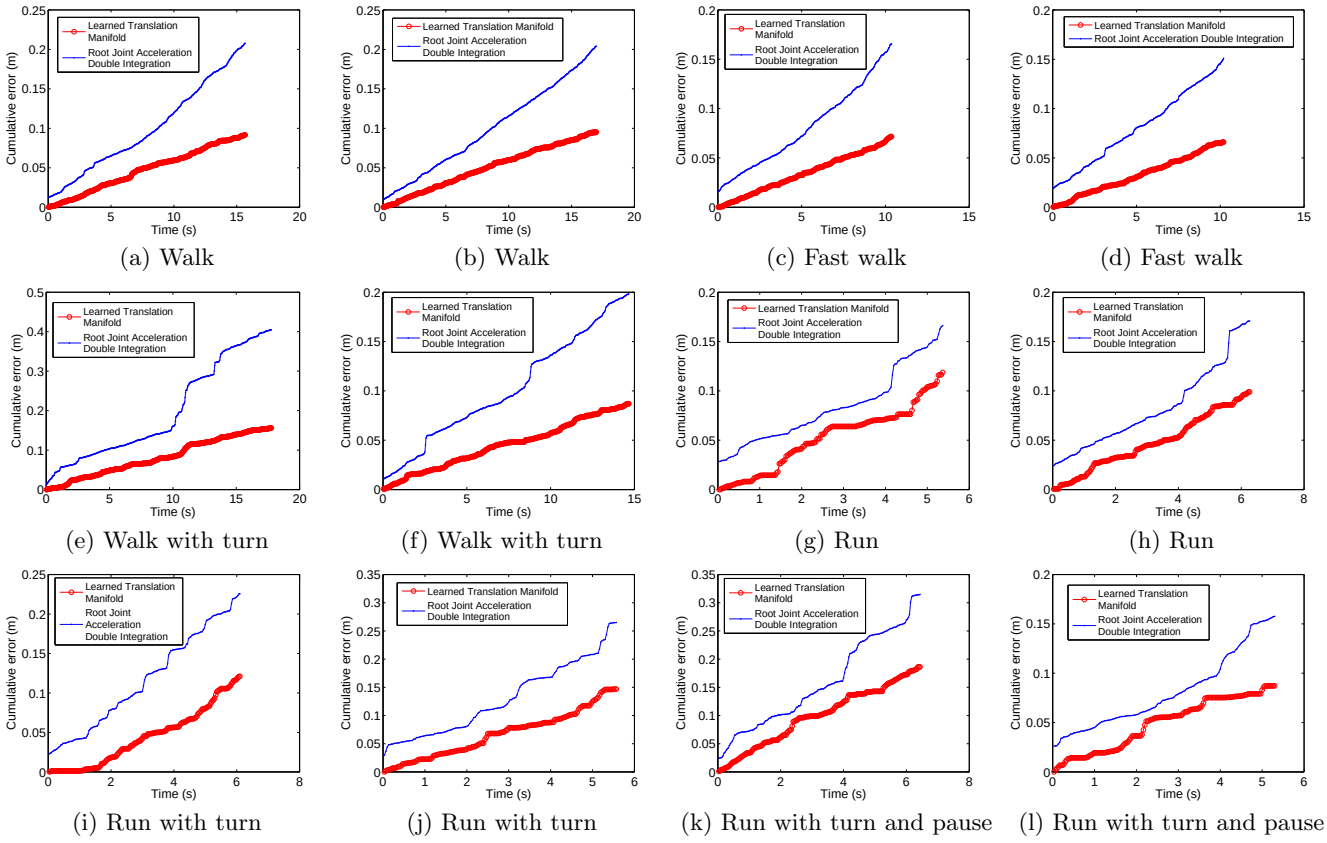
Let $\breve{d\pi}_t$ be the subject's estimated posture variation at time t, and let $\bar{\theta}_t$ be the subject's absolute orientation at that time. Furthermore, let dt_t be the length of the time interval over which $\breve{d\pi}_t$ was recorded. The projection of $\breve{d\pi}_t$ on the translation manifold, $\breve{\phi}_t$, is computed as $\breve{\phi}_t = \breve{d\pi}_t \cdot \mathbf{M}$, where $\mathbf{M}$ is the learned projection mapping from the high-dimensional to the latent low-dimensional space. The cluster nearest to $\breve{\phi}_t$ is given by

$$c^* = \arg\min_{c_i \in \mathbf{C}} \delta(\breve{\phi}_t, c_i.\mu_i) \tag{17}$$

where $\delta(\cdot,\cdot)$ is the Euclidean distance between two points. Then, if $\mathbf{W}^*$ is the cluster translation mapping for c^*, our model predicts a normalised translation for $\breve{\phi}_t$ as

$$\widehat{dp}_t \doteq (\widehat{dx}, \widehat{dy}, \widehat{dz}) = \mathbf{W}^* \cdot \breve{\phi}_t^T \tag{18}$$

The updated predicted position at time t, $\tilde{x}_t, \tilde{y}_t, \tilde{z}_t$, is obtained by applying the orientation $\bar{\theta}_t$ to $\widehat{dp}_t$, scaling it by dt_t



(a) Walk (b) Walk (c) Fast walk (d) Fast walk

(e) Walk with turn (f) Walk with turn (g) Run (h) Run

(i) Run with turn (j) Run with turn (k) Run with turn and pause (l) Run with turn and pause

Figure 6: Cumulative translation errors,12 CMU database motions. *Red:* Learned translation manifold trained on given motion sequence. *Blue:* Double integration of root joint acceleration.

to reflect the length of the current time interval, and adding it to the previously estimated position, $(\tilde{x}_{t-1}, \tilde{y}_{t-1}, \tilde{z}_{t-1})$:

$$\begin{pmatrix} \tilde{x}_t \\ \tilde{y}_t \\ \tilde{z}_t \end{pmatrix} = \begin{pmatrix} \tilde{x}_{t-1} \\ \tilde{y}_{t-1} \\ \tilde{z}_{t-1} \end{pmatrix} + dt_t \begin{pmatrix} \cos(\bar{\theta}_t) & -\sin(\bar{\theta}_t) & 0 \\ \sin(\bar{\theta}_t) & \cos(\bar{\theta}_t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \widehat{dx} \\ \widehat{dx} \\ \widehat{dz} \end{pmatrix}, \tag{19}$$

starting at the position $(\tilde{x}_{t-1}, \tilde{y}_{t-1}, \tilde{z}_{t-1}) = (x_0, y_0, z_0)$.

Our method can generate translations from novel instances of feature vectors, and track the position of a subject without an optical source. This property is important in complex unconstrained environments, where optical systems cannot be directly applied. As joint angles are inherently supplied by the inertial devices, our approach can simultaneously track both position and posture from a ***single*** set of sensors.

4. RESULTS

4.1 Simulation results

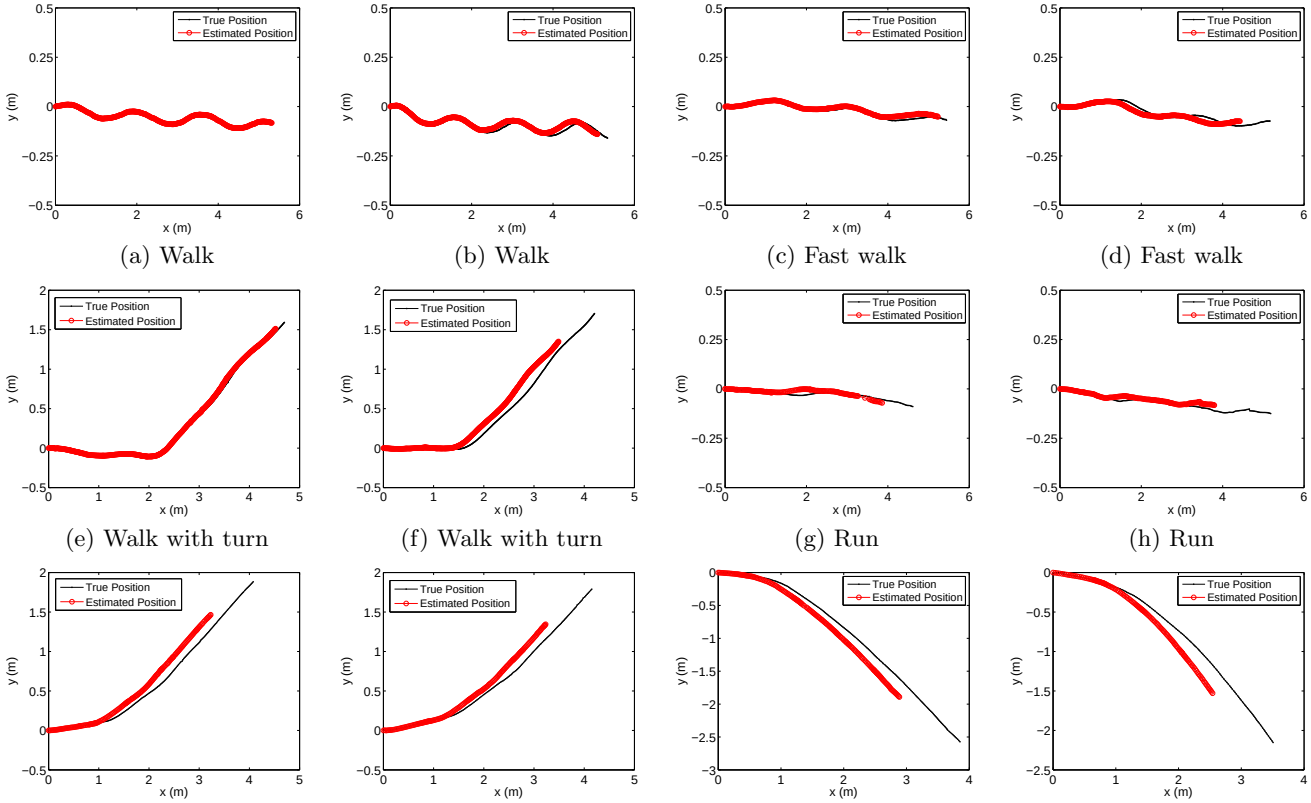
The learning framework was first evaluated on sequences from the Carnegie Mellon University (CMU) Database [3]. Motions in this dataset were captured using an optical system that tracks reflective body markers. Posture vectors were formed by aggregating the marker positions for the lower body joints (thighs, shins, ankles, feet). Note that this is a slightly different representation to the one given in Section 3, where posture vectors consisted of joint angles, not joint positions. However, this differentiation does not impact the applicability of our algorithm, which has no internal model of the nature of the supplied feature vectors.

The position of the *root joint*, at the subject's hips, was taken as the absolute position of the body. This was used as a ground-truth benchmark, against which the iteratively predicted positions were checked.

We compared against a related open-loop position generation technique, the *double integration* of acceleration. This method similarly generates local translations that can be chained together to compute positions. As the CMU dataset does not explicitly provide acceleration data, we simulated this information by extracting accelerations from successive positions at the subject's root joint, and integrating them twice to generate translations.

We first assessed the ability of translation manifolds to reproduce translations on the datasets they are trained on. Towards this end, we trained the learning algorithm on several different motion sequences. We then compared the similarity of the generated translations to the ground-truth translations, as estimated by differences of consecutive root joint positions. Our metric is the *cumulative translation error*, obtained by iteratively summing the Euclidean distance of each generated vector from the corresponding ground truth.

The results for 12 distinct motion sequences, ranging from simple straight walking to running with turns, are shown in Figure 6. In all cases, the translations generated by the



(a) Walk (b) Walk (c) Fast walk (d) Fast walk

(e) Walk with turn (f) Walk with turn (g) Run (h) Run

(i) Run with turn (j) Run with turn (k) Run with turn and pause (l) Run with turn and pause

Figure 7: Overall position error estimation for 12 novel instances of unseen motion sequences. *Red:* Trajectories estimated by learned translation manifold. *Black:* Ground-truth trajectories.

learned manifold yield a lower cumulative error than the corresponding double integration ones. For the simpler walking motions, the discrepancy between the two methods is shown to increase over time, thus suggesting that the translation manifold is more effective at capturing motion dynamics.

The true potential of learned translation manifolds can be fully assessed when applied on *novel* instances of previously unseen motions. In our second simulated experiment, we trained our model on a dataset consisting of 11 different motions by the same subject: a straight walk, two straight walks followed by a 90° left/right turn, two walks with a left/right veer, a fast straight walk, a straight run, two straight runs followed by a 90° left/right turn, and two runs with a left/right veer. The total duration of these captures is 114 seconds, with walking-type and running-type motions accounting for 86 and 28 seconds, respectively. By including different types of motions, our aim was to model a wide range of posture variation-translation pairs, and improve generalisation to novel motion instances.

The learned mapping was applied on 12 new motion sequences of various types. For these motions, we measured the discrepancy between the trajectories predicted by the manifold, and the ground-truth trajectories. The resulting trajectories are demonstrated in Figure 7. As previously, our algorithm is shown to reproduce accurate positions for normal walks, with the error increasing for running-type motions. This increase is partly explained by the larger number of walking motion data points in the training set, which biases the manifold towards translations of smaller magnitude.

4.2 Experimental results

In the physical experiments, we evaluated the translation learning algorithm on sensory data obtained from physical devices, using the Kinect as the optical and the Orient platform as the inertial sensing source. In the training phase, synchronised data from the two sources were used to learn translation manifolds. An important restriction in this case was the small capture volume of the Kinect (approximately 15m^3), which limited the variety of motions that could be performed by the subject. Thus, our framework is evaluated mainly on walking motions which require less physical space.

4.2.1 Constrained environment experiments

The learning algorithm was first compared with the acceleration integration method against ground-truth positions estimated by the Kinect. Unlike simulation experiments, accelerations were now directly supplied by the accelerometers of Orient devices, so translations were generated through double integration of this data.

We captured 18 motion sequences of variable length, ranging from 20 to 180 seconds. We used a total of 4 Orient devices, placed on the subject's waist (root joint), right thigh, left thigh, and left ankle. Motions were captured in an office environment, which impacted the quality of the sensory readings, especially magnetometers, due to metal in

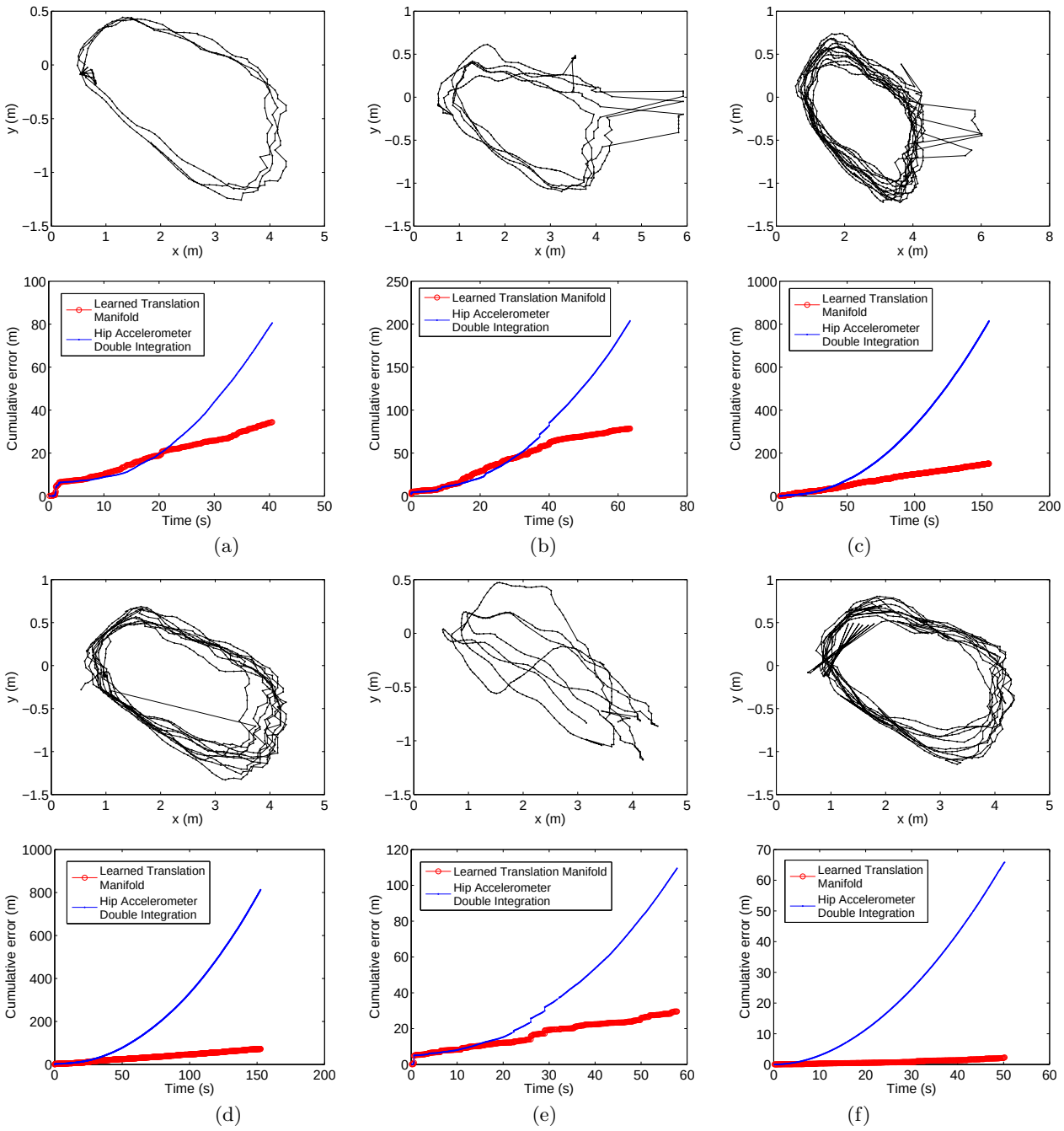
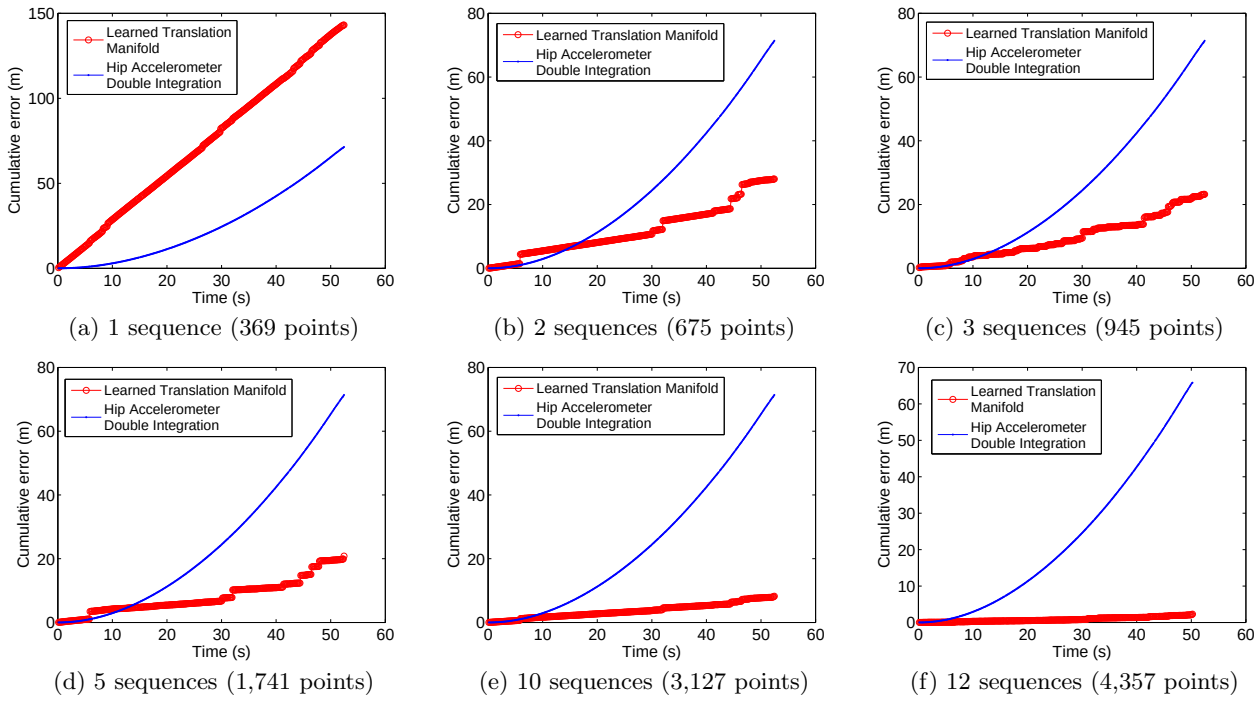


Figure 8: Cumulative generated translation errors for 6 novel motion sequences. The learning algorithm was trained on 12 sequences of varying length and motion composition. *Top of each subfigure:* Ground-truth positions computed by the body tracking interface. *Bottom:* Cumulative errors. *Red:* Learned translation manifold. *Blue:* Double integration of the acceleration of the root joint.

the building structure. For each capture, the subject was allowed to perform any sequence and combination of walking and standing, provided s/he remained within the capture area of the Kinect.

We selected 12 of the captured sequences as the training set, and we used the remaining 6 as novel instances for evaluation. As with simulation experiments, we first compared the cumulative error of translations generated by the learned manifold, and translations from double integration of root joint accelerations.

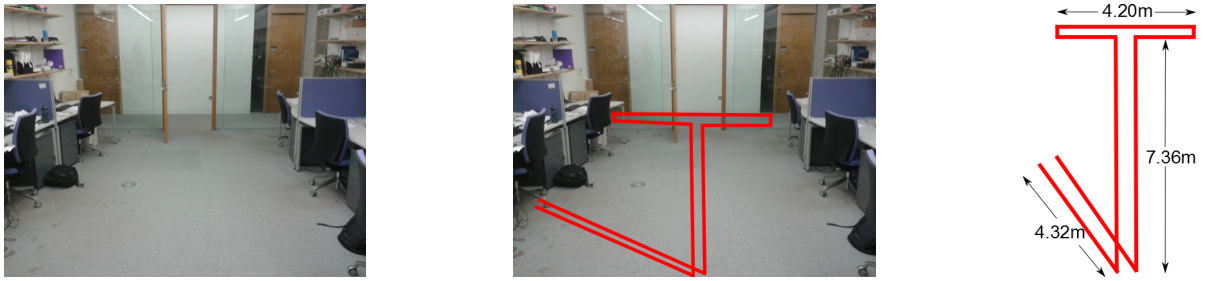
Figure 8 illustrates this comparison, along with the corresponding ground-truth positions captured by the Kinect. In all 6 trials, the subject was observed to repeatedly move around the capture area in a loop. Although in some cases the cumulative error generated by the double integration method was initially lower, in all trials the learning method



(a) 1 sequence (369 points) (b) 2 sequences (675 points) (c) 3 sequences (945 points)

(d) 5 sequences (1,741 points) (e) 10 sequences (3,127 points) (f) 12 sequences (4,357 points)

Figure 9: Effect of training data set size on generated translations. The cumulative error is shown to decrease as the number of training motion sequences (and training data points) increases.



(a) (b) (c)

Figure 10: Unconstrained environment illustration. (a): Office room and corridor. (b): Illustration of the approximate trajectory followed by the subject, starting and ending at the same point. (c): Approximate dimensions of the trajectory.

had a considerably lower error at the end of the sequence. This superior performance was achieved despite some irregularities in the captured positional data, as, for example, in Figures 8(b) and 8(c). This demonstrates that our approach can learn a robust translation model from noisy sensory data, which can yield more accurate translations than methods operating directly on raw data.

The error of the translation manifold algorithm inevitably depends on the size and quality of the training data set. To better understand this effect, we assessed the performance of the algorithm on the last trial of Figure 8 under varying training sets. The results are shown in Figure 9, where we start with just one training sequence, comprising only a few data points, and progressively increase this number. It can be seen that when only one short sequence is supplied, the performance is considerably worse than the double integration method. However, as more motion instances are added to the training set, the error is shown to decrease significantly over time. This indicates that the learning model relies on a good coverage of the posture and translation space, in order to be able to generalise effectively to novel instances. Thus, when recording data, it is important to ensure that the tracked subjects perform a wide range of motions, including various combinations of different motion types (e.g. straight walks and turns).

Another related constraint on the performance of our algorithm is that motions captured during training must be similar to those executed in the online generation phase. For example, if a manifold is learned only from walking motions, it is highly unlikely to yield accurate translations on novel running motions. Thus, it is essential to capture not only a significant *quantity* of data (as shown in Figure 9), but also representative sequences that will be *qualitatively* similar to the motions the system will be tested on when deployed.

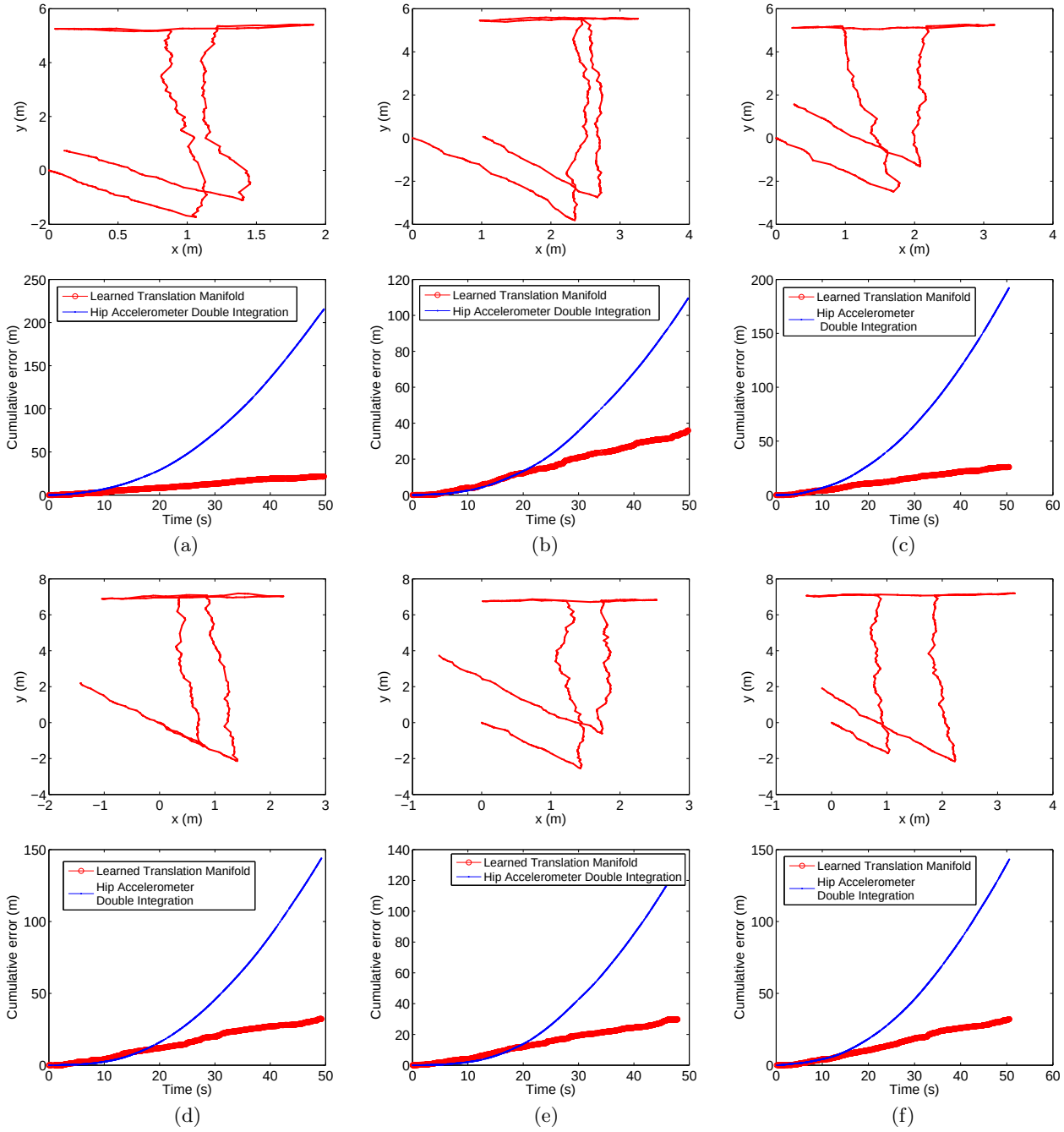


Figure 11: Generated positions for a subject following the trajectory shown in Figure 10, 6 trials. *Top subfigures:* Generated positions. *Bottom:* Cumulative errors and comparison with double integration.

4.2.2 *Unonstrained environment experiments*

In the second set of physical experiments, we evaluated the learning method in an unconstrained office environment (Figure 10). This represents a setting where an optical source cannot be used to track subjects, due to its larger area and morphology (doors, corridors). The subject was asked to follow a trajectory consisting of several landmark points, located inside an office room and in an adjacent corridor. There was no restriction on the time given to follow this trajectory, so the subject was allowed to pause for arbitrary periods of time.

We used the same set of training sequences as in the first set of physical experiments to learn a translation manifold. Figure 11 shows the generated trajectories for six distinct trials of the subject moving along the prescribed path, along with the corresponding error comparison with the double integration method. The true precise trajectory followed in each case was not known, however, the subject always ended his path at the same point where he started. Thus,

by comparing the difference between the start and end points of each generated trajectory, we can get an estimate of the resulting error.

Despite not being aware of the duration and nature of the motions performed by the subject, the learning algorithm is observed to produce translations that closely follow the true trajectory. The computed mean error for the final position was 1.783m, with the overall sum of distances between landmark points being about 30m. Furthermore, as shown in the bottom subfigures of Figure 11, the learning algorithm maintains the superior performance level over the double integration method. A common trait of both sets of physical experiments is that they feature several alternations between straight walking and turning motions, which are characterised by repeated variations in the velocity profile of the tracked subject. In this context, the double integration method initially produces an error comparable with the learning algorithm, but in both cases the margin increases exponentially over time. The learning method is therefore successful in identifying the salient structure of the high-dimensional data, and using it to learn a mapping that can be applied to novel motions.

5. CONCLUSIONS

We have presented a method for simultaneous posture and position tracking in unconstrained environments, based on learned generative translation manifolds. In an offline learning phase, two heterogeneous tracking sources, an inertial sensing (Orient-4) and an optical (Kinect) platform, are jointly used to learn a mapping from posture variations, as estimated by the former, to whole-body translations, as estimated by the latter. This mapping is learned through linear regression on clustered latent representations of posture variations. Online, the optical source is removed, and the learned translation manifold is used to generate translations for novel motion instances. The generative method is experimentally shown to outperform the related model-free, dead-reckoning method of acceleration integration, and to correctly reproduce the structure of previously unseen trajectories in unconstrained environments.

One drawback of our approach is that a different mapping must be learned whenever the system is tested on a new user. This characteristic is due to skeletal morphology and limb dimension constraints, which vary among different subjects. Thus, motion sequences captured on a specific subject may not adequately cover the posture difference and translation space for a different subject, thus leading to incomplete mappings. Nevertheless, one interesting extension to our work would be to learn translation manifolds from datasets which contain motions from various subjects with different characteristics (e.g. short/tall). This extension would be well-suited to the feature vector clustering procedure described in Section 3.2.1. In this context, we would employ a *hierarchical* clustering approach, where the evaluated subject would first be matched to the nearest (in terms of body morphology) user in the training data set, and then a translation would be generated based on the learned mappings for the matched subject.

A major strength of our approach is that it does not make assumptions about the nature of the performed motion, the number and placement of inertial measurement units, or the morphology of the tracked subject's body. This property is advantageous for two reasons. First, our method can be applied to complex motions spanning all three dimensions (e.g. forward jumps), where traditional model-based approaches tracking gait events and foot contacts would fail. Second, for simpler, planar motion types (e.g. walking sequences), our method can be used as a predictive step for model-based filtering approaches, in order to obtain lower positional errors. Extending our work in these directions would further emphasise the benefits of using machine learning techniques to exploit the structure of high-dimensional data produced by physical sensor networks.

Acknowledgments

AV has been supported by a doctoral studentship from the Centre for Speckled Computing, funded by the Scottish Funding Council (Strategic Research Development Programme R32329) and EPSRC (Basic Technology Programme C523881).

6. REFERENCES

[1] Microsoft Kinect. `http://www.xbox.com/kinect`.

[2] OpenNI kinect body tracking interface. `http://www.openni.org`.

[3] CMU motion capture database. `http://mocaps.cs.cmu.edu`.

[4] Vicon motion capture systems. `http://www.vicon.com`.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[6] J. Casper and R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(3):367–385, 2003.

[7] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao. Learning to walk through imitation. In *International Joint Conference on Artifical Intelligence (IJCCAI)*, pages 2084–2090, 2007.

[8] J. A. Corrales, F. A. Candelas, and F. Torres. Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter. In *International Conference on Human-Robot Interaction (HRI)*, pages 193–200, 2008.

[9] R. Feliz, E. Zalama, and J. G. Garcia-Bermejo. Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1):35–42, 2009.

[10] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25(6):38 –46, 2005.

[11] J. Guerrieri, M. Francis, P. Wilson, T. Kos, L. Miller, N. Bryner, D. Stroup, and L. Klein-Berndt. Rfid-assisted indoor localization and communication for first responders. In *European Conference on Antennas and Propagation (EuCAP)*, pages 1–6, 2006.

[12] H. Khoury and V. Kamat. Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, 18(4):444–457, 2009.

[13] Y. Kobayashi and Y. Kuno. People tracking using integrated sensors for human robot interaction. In *IEEE International Conference on Industrial Technology*, pages 1617 –1622, 2010.

[14] K. F. MacDorman, R. Chalodhorn, and M. Asada. Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation. In *International Conference on Pattern Recognition (ICPR) (4)*, pages 537–540, 2004.

[15] L. Ojeda and J. Borenstein. Personal dead-reckoning system for gps-denied environments. In *IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1 –6, 2007.

[16] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, 2011.

[17] A. Valtazanos, D. K. Arvind, and S. Ramamoorthy. Comparative study of segmentation of periodic motion data for mobile gait analysis. In *ACM International Conference on Wireless Health*, pages 145–154, 2010.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[19] A. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari. Distributed segmentation and classification of human actions using a wearable motion sensor network. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1 –8, 2008.

[20] A. D. Young. From posture to motion: the challenge for real time wireless inertial motion capture. In *International Conference on Body Area Networks (BodyNets)*, pages 131–137, 2010.

[21] A. D. Young, M. J. Ling, and D. K. Arvind. *Orient-2*: a realtime wireless posture tracking system using local orientation estimation. In *Workshop on Embedded Sensor Networks (EmNets)*, pages 53–57, 2007.

[22] X. Yun, E. Bachmann, H. Moore, and J. Calusdian. Self-contained position tracking of human movement using small inertial/magnetic sensor modules. In *International Conference on Robotics and Automation (ICRA)*, pages 2526 –2533, 2007.