

Article

Smart Annotation of Cyclic Data Using Hierarchical Hidden Markov Models

Christine F. Martindale ^{1*} , Florian Hoenig ², Christina Strohrmann ³ and Bjoern M. Eskofier ¹ 

¹ Machine Learning and Data Analytics Lab, Department of Computer Science, Friedrich-Alexander University Erlangen-Nürnberg (FAU), 91054 Erlangen, Germany; bjoern.eskofier@fau.de

² Speech Group, Department of Computer Science, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), 91054 Erlangen, Germany; florian.hoenig@cs.fau.de

³ Bosch Sensortec GmbH, Gerhard-Kindler-Strasse 9, 72770 Reutlingen, Germany; annachristina.strohrmann@bosch-sensortec.com

* Correspondence: christine.f.martindale@fau.de; Tel.: +49-9131-85-20159

Received: 6 September 2017; Accepted: 11 October 2017; Published: 13 October 2017

Abstract: Cyclic signals are an intrinsic part of daily life, such as human motion and heart activity. The detailed analysis of them is important for clinical applications such as pathological gait analysis and for sports applications such as performance analysis. Labeled training data for algorithms that analyze these cyclic data come at a high annotation cost due to only limited annotations available under laboratory conditions or requiring manual segmentation of the data under less restricted conditions. This paper presents a smart annotation method that reduces this cost of labeling for sensor-based data, which is applicable to data collected outside of strict laboratory conditions. The method uses semi-supervised learning of sections of cyclic data with a known cycle number. A hierarchical hidden Markov model (hHMM) is used, achieving a mean absolute error of 0.041 ± 0.020 s relative to a manually-annotated reference. The resulting model was also used to simultaneously segment and classify continuous, ‘in the wild’ data, demonstrating the applicability of using hHMM, trained on limited data sections, to label a complete dataset. This technique achieved comparable results to its fully-supervised equivalent. Our semi-supervised method has the significant advantage of reduced annotation cost. Furthermore, it reduces the opportunity for human error in the labeling process normally required for training of segmentation algorithms. It also lowers the annotation cost of training a model capable of continuous monitoring of cycle characteristics such as those employed to analyze the progress of movement disorders or analysis of running technique.

Keywords: hierarchical hidden Markov models; segmentation; smart annotation; cyclic sensor data; semi-supervised learning; annotation cost; activity recognition; gait classification; inertial sensors; wearable sensors

1. Introduction

Cyclic movements intrinsic to daily life range from human motion such as running, cycling and rowing to biological signals such as the electrical activity of the heart. Understanding and quantifying these cyclic data is especially important for the analysis of movement disorders, sports, rehabilitation and general health [1–3]. The first step in the analysis of these cyclic data is the segmentation of them into individual cycles for direct analysis such as temporal parameter calculation (e.g., cycle time) or as training data for more complex algorithms based on pre-segmented data (e.g., stride length). Currently, there is no data processing pipeline, applicable to the movement analysis or biological signal domain, capable of unsupervised cyclic event segmentation of sensor-based data. Therefore, time-consuming manual annotation is currently required to achieve this type of segmentation, even just for validation datasets. A generic segmentation pipeline that could segment any arbitrary, previously

unseen, periodic data would be valuable to these applications. The application of the pipeline would not be limited to periodic human movement data, but could be applied to any cyclic data.

Completely automatic annotation of cycles within sensor data is essentially an unsupervised segmentation task. However, the current state of the art in sensor-based systems falls mainly in the supervised domain. Segmentation algorithms can be split into three main categories of training methods: supervised, semi-supervised and unsupervised. In the supervised domain, there is an abundance of different segmentation algorithms such as dynamic time warping (DTW), longest common subsequence (LCSS) and thresholding with windowed peak detection (WPD) [4]. The main disadvantage of these supervised techniques is the annotation cost involved because each cycle in the training set must be manually segmented in order to calculate a template or threshold [5]. Reduction of this annotation cost for gait data has been addressed by using instrumented treadmills or mats [6] or by using shoes with pressure-sensitive insoles [7]. In both cases, the thresholded pressure signal is used to determine stride boundaries. However, both are restricted by the availability of such large or expensive devices, which lack in-the-field applicability. This method of automated annotation applies only to gait data.

In the semi-supervised domain, there have been several methods proposed to reduce the amount of fully-labeled data required. One of the more popular solutions is active learning or labeling [8,9]. This method identifies which sections of data would be most effective to annotate. A variation on this theme is to split the data into similar ‘populations’ and then fully label only a subsection of each population [10–12]. Alternatively, any shared structure between activities can be exploited [13]. These techniques have been applied only on a high level, for example to reduce the annotation cost when classifying and segmenting whole activities, however not when also segmenting these activities into individual cycles. These techniques are semi-supervised because they require labeled data, but less than for their supervised counterparts, which required fully-labeled data.

In the completely unsupervised domain, sometimes called ‘zero-resource’ because no prior information is given, there is even less literature. Within the activity recognition domain, there are some unsupervised methods, which enable segmentation of primitives within activities; however, these primitives are also found in non-cyclic data, and their boundaries vary per person, activity and method [14,15]. These methods have been demonstrated on motion capture data only on the activity level. Some online, unsupervised activity recognition methods exist [16], some of which are based on exploiting the hierarchical structure of motion data [17,18], although these do not identify primitives within each activity. Within a smart home setting, unsupervised methods have been used for classification of interleaved activities [19].

However, to the best of the authors’ knowledge, an unsupervised segmentation algorithm for both classification of activities and for segmentation of those activities into cycles has not been achieved within this domain. We focus on a method applicable to limited sensors, specifically one wearable IMU. The proposed method does not rely on a model of the human body, such that it could also be applied to other cyclic data as it relies only on the periodicity of the data. Ultimately, zero-resource techniques would completely remove the need for any level of annotation for training purposes.

This paper investigates the first steps required to achieve the aim of unsupervised segmentation for cyclic activities, without using an explicit body model. The advantage of this is that it can be used with a low number of sensors and is independent of sensor location. We present a method, based on a hierarchical hidden Markov model (hHMM), which enables the semi-supervised segmentation of uninterrupted cyclic data with a given repetition count. Due to the proposed model requiring uninterrupted cyclic data and the repetition count, it falls into the semi-supervised domain although it differs from the existing semi-supervised techniques because it requires no fully-labeled data sections.

The feasibility of this method is demonstrated using continuous walking and running data collected from a body-worn sensor. The results are compared to manual segmentation by a human. The generated model, trained in a semi-supervised manner, is then applied to simultaneous

segmentation and classification of previously unseen data. The results of a fully-supervised version are used as a baseline.

2. Background

In order to alleviate the level of supervision required, i.e., to implement a smart annotation method, a model-based algorithm, rather than a template-based one, must be chosen. This is because a template-driven approach requires fully-labeled training data in order to form a template. However, a model-based approach can progressively learn a model by an iterative process of learning and relabeling the input data. A family of models that meet these criteria are graphical models such as hidden Markov models (HMM), conditional random fields (CRF) and semi-Markov models (SMM). Thomas et al. (2010) investigated semi-supervised SMMs for simultaneous segmentation and classification. However, their method requires some fully-labeled data, as well as prior expert knowledge of the likely sequence of events [20]. The rhythmic extended Kalman filter has also been used for unsupervised gait analysis; however, it relies on a body model, assumes walking in a straight line and requires multiple sensors firmly attached to specific locations on the subject, so although it is online and unsupervised, it requires patient-specific information and may not be extensible to other cyclic movements. Furthermore, the segmentation results were only qualitatively compared [21,22]. Markov chain Monte Carlo was also used for unsupervised activity recognition; however, individual cycles within activities were not segmented [18].

HMMs are a well-understood model for sequential data, using Gaussian mixture models (GMMs) to represent the data [23,24]. Mannini et al. (2015) investigated a fully-supervised HMM for gait segmentation and Andreato et al. (2006) for ECG segmentation [6,24]. However, a four-state, circular HMM, as used by Mannini et al., does not model the transition between different classes of signal such as standing to walking [6]. Andreato et al. used expert knowledge to model the transitions within the internal structure of an ECG signal [24]. However, a more intuitive and efficient approach to address the underlying structure of cyclic data and to provide a model for the transition between different classes of a cycle is a hierarchical HMM (hHMM) [25]. The hierarchical structure allows for more efficient methods of estimating parameters and of training the individual HMMs. This hierarchical concept allows a model for each class of data and a transition model between individual HMMs (Figure 1). The above-mentioned models were all implemented in a fully-supervised manner and therefore require the entire dataset to be labeled on a per cycle basis. HMMs have been used for unsupervised, online, activity recognition and segmentation [16,17]; however, they did not further segment activities into cycles. The use of hierarchy combined with HMM has been used by both Lv et al. and Kulic et al.; however, [26] use HMMs as a weak classifier for AdaBoost, so using hierarchy, and [17] used a tree classification approach. HMMs in this field have also been used in a semi-supervised approach where the segments are initially found using velocity peaks and zero crossings, although segmentation accuracy was only analyzed for within 0.2 s [27]. This method relies on a specific shape and thresholds to perform in a semi-supervised manner.

These hHMMs have been widely used in speech recognizers [28], as well as for fully-supervised segmentation and classification of human movement data [29]. In the context of speech recognizers, the hHMM iteratively learns both the event models and the event segmentation. A similar method is applied in this paper, for application to cyclic data. However, cyclic data are made up of repetitions of the same event, whereas speech data rarely contain repeated words or phonemes, thereby reducing the ambiguity of the task. Therefore, this paper tackles their application, in a semi-supervised manner, to cyclic movement data rather than speech data. The method proposed in this paper differs from previous semi-supervised and unsupervised approaches in that it is applicable to data from a single sensor and does not inherently rely on knowledge of the location of the device, type of device or the shape of the signal. This enables its application to a single wearable device rather than specifically to motion capture data or those relying on a model of the body and at least five sensors. It also enables the

influencing of the segmentation boundary based on start and end points of cycles within the training data, while also allowing these points to be arbitrary, depending on the application of the method.

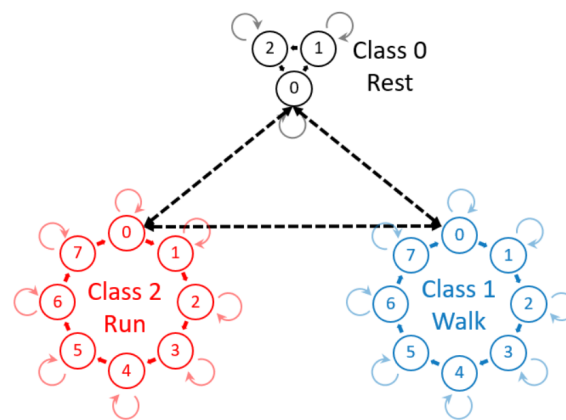


Figure 1. Hierarchical hidden Markov model (hHMM) consisting of three classes each of, respectively, 3, 8 and 8 internal states. The internal states are strictly circular with the possibility of recursion. The dashed arrows represent the transition model between individual HMMs that represent classes such as rest, walking and running.

3. Proposed Pipeline

3.1. Smart Annotation

The basis of this method is the concept of iterative learning from an initial estimate: the linear division of a data section by the cycle repetition number (Figure 2). As input, the pipeline requires sections of data containing one class of cycle and the number of repetitions within that section. With this information, the data are linearly divided into the given number of repetitions (Box 1, Figure 2). The resulting naive, uniform segmentation is then used for semi-supervised training of an HMM using Viterbi training [30], i.e., an efficient parameter learning alternative to Baum–Welch (Box 2, Figure 2). The training of the internal states is also initiated in a linear manner, hence making it also semi-supervised. These linear segmentations are then used to train an HMM, and the number of training iterations performed is T_i . This limit can be found empirically or set very high, if computation time is not a restriction. The resulting, trained HMM is then used to compute a forced alignment (Box 3, Figure 2). Forced alignment is the calculation of the Viterbi path to find the most probable state sequence given a known sequence of events and is commonly used and implemented within speech processing toolkits [31]. Forced alignment, in this case, simply calculates the most probable Viterbi path where the resulting path must contain the defined number of iterations. This is achieved by concatenating the given sequence, i.e., it is forced to only estimate sequences that result in the given number of repetitions. This results in new segmentation boundaries. The results of this forced alignment, the new segmentation boundaries, are then used as supervised training for the next iteration (Box 2, Figure 2). From this point on, it is a supervised model where the segmentation boundaries resulting from the forced alignment are used to split the data for training the HMM. The result of the final forced alignment is then used to calculate the segmentation error of the smart annotation model's segmentation relative to the manually-segmented reference. The desired cycle boundary is found based on the assumption that each segment of data (primitive containing known cycle number) for training contains a known whole number of cycles, and hence, the stride boundaries will be the beginning and end of a cycle.

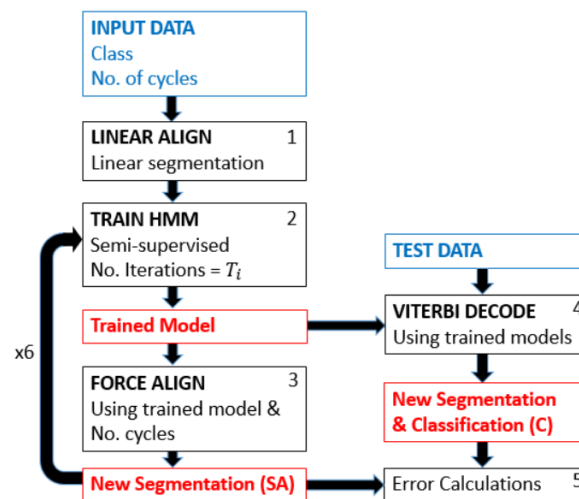


Figure 2. Pipeline showing the smart annotation method on the left and the use of the resulting models for simultaneous classification and segmentation of test data on the right.

In order to avoid overfitting to an incorrectly segmented model (the one learned with the uniformly-segmented data), the number of training iterations, T_i , is gradually increased. For the implementation used in this paper [$T_i = 1, 2, 5, 8, 13, 21$], and hence, the method looped six times between HMM training (Box 2, Figure 2) and forced alignment (Box 3, Figure 2). The number of loops was limited in this manner due to only minimal improvement with more iterations and was fixed for all experiments in this paper.

The processing of all input data segments (primitives containing known cycle number) is calculated simultaneously, preventing overfitting, as well as allowing for faster processing than running the pipeline for individual samples. This means that the models are learned simultaneously. Any number of models can be learned simultaneously; in fact, a larger variety of data to be learned simultaneously enables the model to better resolve ambiguities found when learning a model.

This pipeline, the combination of linear alignment, HMM training and forced alignment, is in this paper termed smart annotation (SA) because the resulting segmentation of the forced alignment can be used as annotations for the input data with limited supervision (repetition number and class).

3.2. Simultaneous Smart Annotation and Classification

A further application of this final SA model is simultaneous segmentation and classification of continuous data, with no further training than the smart annotation described. This application refers to the use of the hHMM trained using the SA method to classify and segment unseen, continuous data where no cycle numbers or segments are known. However, an additional HMM must be trained to recognize the non-movement sections that naturally occur in continuous data. This is a similar step to the simultaneous classification and segmentation carried out by [29]; however, in our implementation, the model is semi-supervised, as opposed to their fully-supervised version. If the test data are a mixture of classes, then each class present would need an HMM trained by the smart annotation method, and the transition probability between these models can be set to equal, i.e., naively initiated and not trained. With these models and the equal transition probabilities, Viterbi decoding can be used to estimate the segmentation, as well as classification of unseen test data (Box 4, Figure 2). The calculation of the Viterbi path here uses a fully-connected transition model, such as that in Figure 1. This means that no simplifications are made due to a known sequence of events, and the model has less prior knowledge than in the previous case, i.e., neither the number of events, nor the sequence.

4. Dataset

In order to investigate the effectiveness of this smart annotation method, a dataset of cyclic activities, in this case walking and running, was collected using an inertial measurement unit (IMU) within the Bosch development platform, which recorded acceleration (± 8 g) and the rate of change of the angle (2000 dps) with a frequency of 200 Hz (dataset available at www.mad.tf.fau.de/research/activitynet). Eighteen participants, 14 male and 4 female, with an average age of 28.3 ± 3.0 years and an average height of 177.1 ± 8.3 cm, performed a sequence of standing, walking and running while wearing the above-mentioned sensor above their left ankle (Figure 3). This paper aims to provide a method applicable to any cyclic data, including where just one sensor is available. The data were collected ‘in the wild’ in that it was an outdoor setting with varying surfaces, with a prescribed circuit, but self-selected speeds, and the transition points were landmarks. The data collection was continuous per person. This was simultaneously video recorded (30 Hz) for annotation purposes. Reference labels were found by simultaneous analysis of video and sensor signal as illustrated in Figure 4. All subjects gave written informed consent, prior to the data collection, in accordance with the ethical committee of the medical faculty at the university.

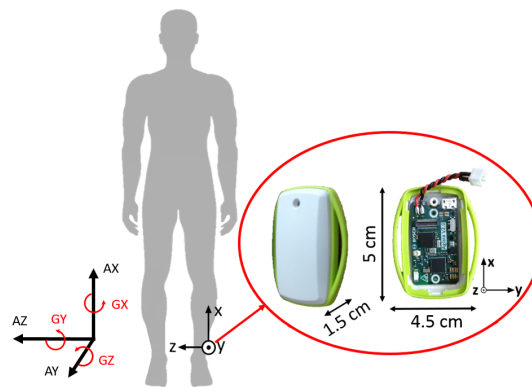


Figure 3. Sensor placement, size and orientation. Three-dimensional acceleration (AX, AY, AZ) and three-dimensional gyroscope (GX, GY, GZ).

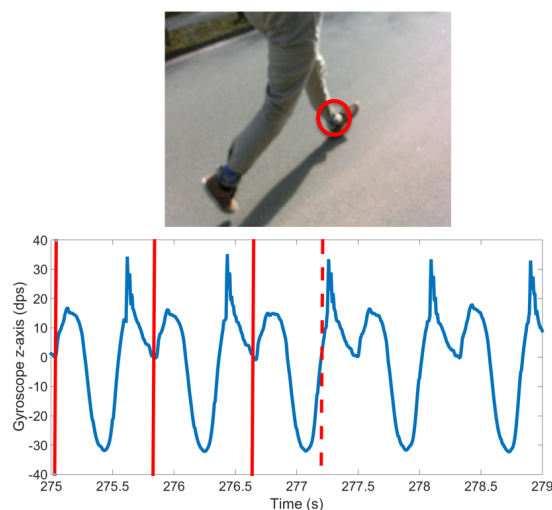


Figure 4. Image of manual labeling tool showing that the signal is the gyroscope sagittal plane (GZ, blue line) of the left foot. Solid vertical lines represent the already annotated cycles. The dashed vertical line corresponds to the shown camera frame. The circle in video frame shows the sensor location.

A cycle, in this case a stride, was labeled such that the beginning of the first stride and end of the last stride in a sequence are data points corresponding to the sensor being stationary. In the case of standing this means that the sensor was vertically above the ankle. Therefore, each stride border (SB) within a bout of walking or running was labeled as the point where the sensor was vertically above the ankle. An example sensor signal with SB labels and classes is shown in Figure 5. This definition enables the precise detection and definition of the beginning of an activity, i.e., the initial movement of the sensor. It also enables the labeling and counting of a low number of stride events. For example, if a subject takes one stride, from standing on the left foot, then there will only be one heel strike of the left foot. If a stride were defined as heel strike to heel strike, then this single stride would not be segmented or classified as a stride. Segmenting strides using the mid-stance or stationary section of a stride has two main benefits: it can immediately be used in algorithms that rely on identification of the zero velocity phase, and it allows for easy isolation of sections of whole cycles as one may just find standing phases and then count the cycles between.

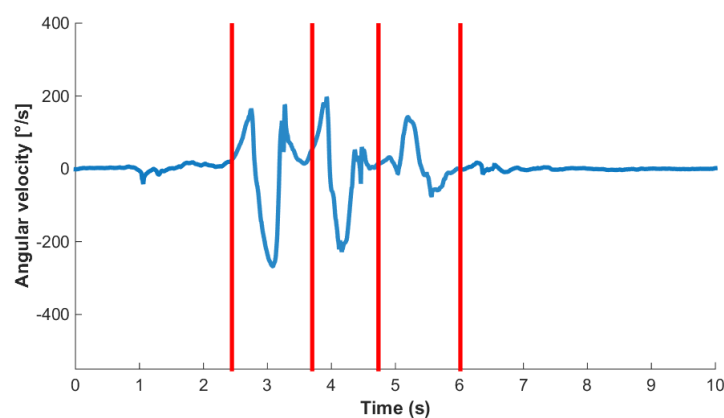


Figure 5. Sample gyroscope signal (blue line) consisting of three strides bounded by rest. Segment boundaries (SB) are shown as vertical lines.

Human annotators, familiar with data from human movement both as IMU data and as camera data, manually segmented all strides and labeled each stride as walking or running. Each labeler annotated an independent section of the data. All instances of standing were labeled as rest. Any section of data that was not within these three categories was labeled as transition and ignored for evaluation purposes. Each labeler annotated an independent section of the data. The IMU data and video were synchronized using a repetitive movement of the sensor at the beginning and end of each recording. A total of 2263 walking strides and 1391 running strides were manually labeled with a minimum of two separate examples of rest (standing) per person.

5. Implementation Details

5.1. Smart Annotation

The processing pipeline (Figure 2) was implemented using the Java Speech Toolkit (JSTK) [32]. The number of internal states per HMM was chosen via a grid search (Table 1). The values for the grid search were chosen partially empirically and partially based on the literature [5,23,33]. The range of internal states used was chosen on the assumption that these numbers of internal states are meaningful for gait, in that gait is often divided into four or eight phases [33]. Ten states were also included to investigate if a higher number of states than is physically meaningful might also be useful. The number of centers for the initialization of the Gaussian mixture models (GMM) used to model the data was trivially initialized to 10. The densities of these were calculated using 10 estimation maximization (EM) iterations. A diagonal covariance matrix was used.

Table 1. Rough grid search for hHMM parameter optimization.

Type of Parameter	Parameter Value
Number of states per HMM	4, 8, 10
Sliding window length (s)	0.25, 0.50, 1.00
Axes combinations (See Figure 3)	AXGZ, GYGZ, AXAYAZGXGYGZ, AYGZ, GZ

5.2. Simultaneous Smart Annotation and Classification

A further application of this hHMM is simultaneous segmentation and classification of continuous, ‘in the wild’ data, with no further training than the smart annotation described. The number of different classes of cycles was two in this case: walking and running. For this part of the pipeline, an HMM representing rest or standing was also trained, due to the test data including sections of rest. The number of internal states for rest was fixed to three. In order to estimate the probable sequence of events of a test signal, i.e., calculate the Viterbi path for a signal, the probability of occurrence of each class is also needed. In order to keep the solution general, all possible events were given an equal probability of occurring, and only the current event was taken into account.

For comparison, an hHMM supervised with the SB, but not the internal states of each stride, was trained and applied to simultaneous segmentation and classification of the same test data.

5.3. Data Preparation

A variety of combinations of axes was chosen to compute features with, and the best was selected via a grid search (Table 1). The gyroscope sagittal plane data featured in all combinations due to its common use in mobile gait analysis [5]. The features chosen were variance, the first three coefficients of the second order polynomial fit, as well as the raw data. The features were calculated from the unfiltered, but calibrated data. These were calculated using a sliding window approach where the overlap was one sample less than the window size. These window sizes were chosen up to the average cycle length (Table 1). A simple feature set was chosen to illustrate the general applicability of this method, as well as to reduce processing time. The coefficients of the polynomial fit were chosen as they describe the shape of the data within a relatively large window with minimal features (e.g., a window of 100 samples needs only three values to describe its shape). All features were normalized, per person, to remove inter person differences in amplitude and mean. It is assumed that this normalization, for application purposes, could be calculated from just a short running trial or estimated based on the height and weight of the user. It also allows the method to potentially be used based on the cyclic nature of the data and not on the placement of the sensor.

5.4. Evaluation Metrics

5.4.1. Smart Annotation Pipeline

In order to assess the segmentation accuracy of the presented pipeline, the mean absolute error in segmentation of the SB was calculated. This is the difference between each estimated SB location and that found by the human annotators. The mean absolute error, on a per cycle basis, of each cycle duration (stride time) was also calculated.

5.4.2. Simultaneous Smart Annotation and Classification

For the evaluation of the effectiveness of classification, the F1-score, sensitivity and specificity were calculated, as well as the confusion matrix. The segmentation evaluation metrics mentioned above are also reported. In this case, each estimated cycle is matched to the corresponding ground truth cycle by finding the two cycles with the maximum overlap. If two steps are estimated where in the ground truth there is only one, then the estimated step with the greatest overlap is used for segmentation and cycle duration error calculation. If an estimated step is two steps in the ground truth, then again, the ground truth step with maximum overlap to the estimated step is used for error

calculation. This erroneous segmentation boundary will be reflected in a very high cycle duration error due to the fact that all cycles are contiguous. This allows the segmentation and cycle duration errors to reflect the accuracy in segmentation when a step is correctly identified. Any false positives or false negatives will be reflected by the classification accuracy results. The error and classification accuracies are calculated for each test person and then averaged per class, on a per sample basis. Samples where the human annotator was unsure of the class, as described previously, were disregarded.

5.4.3. Train and Test Sets

The data from 12 randomly chosen subjects (training set) were used for training both the smart annotation and the simultaneous segmentation and classification models. The data were split randomly into sequences of 5–20 consecutive cycles. The same sequences were used for supervised and smart annotation training purposes, where for supervised training, the SB were given, and for smart annotation, only the repetition number was given. The longer the sequence that the HMM has to learn, the less likely it is to learn a good model. Therefore, the range of 5–20 cycles was chosen as a compromise between lowering annotation cost and keeping the sequence short enough for the HMM to be able to resolve ambiguities in the model.

The remaining six subjects (test set) were used for testing and comparison of the models. This test data were not split, but remained a continuous section of data containing rest, walking and running.

A model was trained for all combinations of the parameter choices in Table 1. In the case of smart annotation, the model that achieved the lowest segmentation error was used for the calculation of evaluation metrics. This model was then also applied to the simultaneous segmentation and classification task. This is the best model choice when segmentation of the input sequences is the only outcome. However, this may not be the optimal model when inter-class transitions are included.

In the case of supervised simultaneous segmentation and classification, the best model was chosen via a three-fold cross-validation of the training set, using the highest F1-score as the criteria for the best model. No subject was both in the training and test set within the cross-validation folds. Using these model parameters from the best model, the whole of the training set (training set) was then used to train a new model. The resulting model was then applied to the test set. For fair comparison of the effect of using smart annotation as opposed to supervised input to the model, the smart annotation model with identical parameters to the best supervised model was also applied to the test set and evaluated.

The same train-test split (12:6) was used for the supervised and semi-supervised methods, and parameter choice was performed via three-fold cross-validation of the training set, ensuring that no data from a subject used for training was used for testing, showing the generalizability of the model.

6. Results

6.1. Smart Annotation Pipeline

The best model for smart annotation was achieved using a window size of 0.5 s, the sagittal plane rate of change of angle (GZ) and four internal states per class. We will refer to this model as SA-GZ4 (Table 2). The mean absolute error of the segmentation of SB, relative to the manually segmented reference, was 0.041 ± 0.020 s, and the mean absolute cycle duration (stride time) was 0.032 ± 0.005 s (Table 2).

The best model according to the fully-supervised hHMM (see the following section) was applied to smart annotation alone (SA-AXGZ8, Table 2). The results are slightly worse, achieving a segmentation error of 0.054 ± 0.010 s, and the mean cycle duration error was 0.035 s, almost identical to that of the SA-GZ4 model, however with a worse standard deviation of 0.055 s.

While small differences were found using the grid search, the overall results did not vary drastically within the grid search with the mean segmentation error being 0.055 ± 0.010 s and the

mean stride duration error being 0.040 ± 0.005 s. This implies that the parameter choice, while helpful, is also fairly robust to non-optimal selection.

Table 2. Mean absolute error in segmentation of stride borders (SB) relative to manual segmentation and mean absolute error in cycle duration (stride time).

Method	Model	Segmentation Error	Cycle Duration Error
Smart annotation alone	SA-GZ4	0.041 ± 0.020 s	0.032 ± 0.005 s
Smart annotation alone	SA-AXGZ8	0.054 ± 0.010 s	0.035 ± 0.055 s
Segmentation & classification, using SA	SA-GZ4	0.101 ± 0.069 s	0.098 ± 0.088 s
Segmentation & classification, using SA	SA-AXGZ8	0.063 ± 0.029 s	0.056 ± 0.025 s
Supervised segmentation & classification	Sup-AXGZ8	0.069 ± 0.029 s	0.064 ± 0.014 s

6.2. Simultaneous Smart Annotation and Classification

6.2.1. Classification and Confusion

The model found to be the best for the smart annotation (SA-GZ4) was then applied to the task of simultaneous segmentation and classification achieving an overall F1-score of 93.2% (Table 3). The F1-score for rest was naturally high, due to its simplicity; however, the F1-scores for walking and running were relatively low at roughly 90% each. The confusion matrix shows some misclassification between walking and running (Table 4).

To give a baseline for comparison, a supervised model was used to segment and classify the test data. The best model choice for this was the gyroscope sagittal plane (GZ) and the vertical acceleration plane (AX), with a window size of 0.5 s and eight internal states (Sup-AXGZ8). This model achieved an overall F1-score of 96.8% (Table 5) and similar misclassifications between walking and running (Table 6).

The best model choice according to the supervised model was then also applied when the initial model was trained in the smart annotation method (SA-AXGZ8). The classification and confusion results are shown in Tables 7 and 8. The results are very similar to the fully-supervised version, sometimes even achieving better results such as for the walking F1-score. An example of the segmentation results on an internal state level can be seen in Figure 6.

6.2.2. Segmentation Error

The segmentation errors for the above-mentioned models show that the segmentation achieved for the AXGZ8 model is comparable for the supervised and smart annotation approaches, while the segmentation error achieved for the model chosen via the best segmentation error for the training set (SA-GZ4) achieved considerably higher errors (Table 2).

Table 3. Classification results, using the smart annotation model (SA-GZ4), on continuous test data on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Overall	Rest	Walking	Running
Sensitivity	91.8	99.4	86.6	89.5
Specificity	98.8	99.2	99.2	97.9
F1-score	93.2	98.5	90.7	90.5

Table 4. Confusion matrix, using the smart annotation model (SA-GZ4), of continuous data including walking, running and rest on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Classified Rest	Classified Walking	Classified Running
True Rest	99.8	0.2	0.0
True Walking	1.1	96.7	2.0
True Running	0.0	1.5	98.5

Table 5. Classification results, using the supervised model (Sup-AXGZ8) on continuous test data on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Overall	Rest	Walking	Running
Sensitivity	96.6	98.7	93.1	97.9
Specificity	99.1	99.7	98.8	98.7
F1-score	96.8	98.8	95.0	96.5

Table 6. Confusion matrix, using the supervised model (Sup-AXGZ8), of continuous data including walking, running and rest on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Classified Rest	Classified Walking	Classified Running
True Rest	99.1	0.9	0.0
True Walking	0.5	98.1	1.4
True Running	0.0	1.1	98.9

Table 7. Classification results, using the smart annotation model (SA-AXGZ8), on the continuous test data on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Overall	Rest	Walking	Running
Sensitivity	95.6	98.7	91.3	96.9
Specificity	99.2	99.5	99.4	98.6
F1-score	96.2	98.6	94.3	95.8

Table 8. Confusion matrix, using the smart annotation model (SA-AXGZ8), of continuous data including walking, running and rest on a per sample basis. All calculated per person and then per class and displayed as a percentage.

	Classified Rest	Classified Walking	Classified Running
True Rest	99.5	0.5	0.0
True Walking	0.7	97.8	1.5
True Running	0.0	0.4	99.6

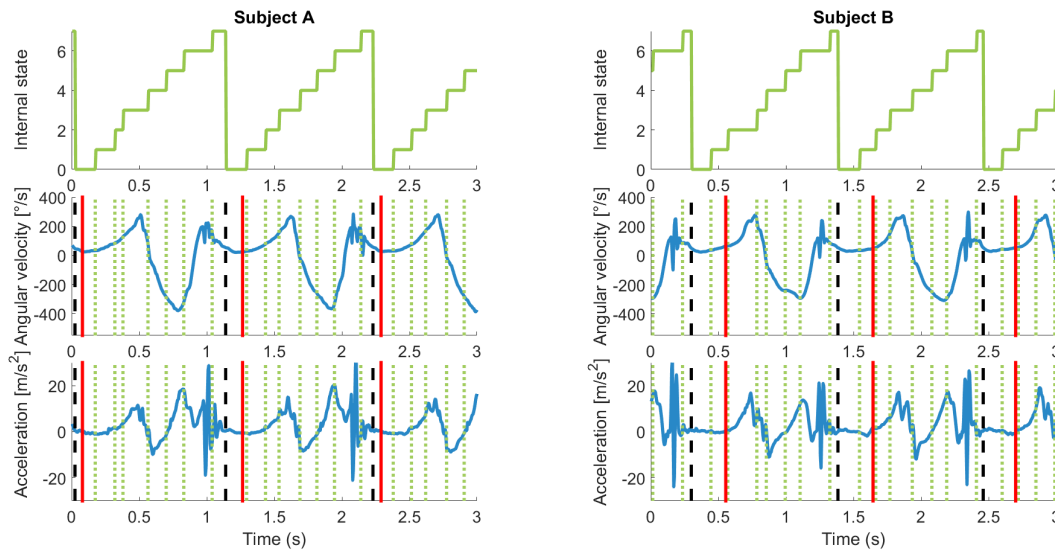


Figure 6. Example of estimated and ground truth labels for two subjects. Left column, Subject A; right column, Subject B. **Top row**, internal states of hHMM with eight internal states. **Middle row**, gyroscope sagittal plane signal. **Bottom row**, vertical acceleration plane. The vertical dotted lines correspond to the internal states. The dashed vertical lines correspond to the estimated start of each cycle. The vertical solid lines refer to the reference labels. The reference labels are inconsistent: for Subject A, they appear at a local minimum and for Subject B not at the local minimum. Some internal states seem to correspond to meaningful data points such as minimums and maximums.

7. Discussion

The results for the smart annotation pipeline show that given a known number of iterations, cyclic data can be segmented, with an absolute segmentation error of 0.041 ± 0.020 s, which is on average 4.0% of an individual cycle duration. This is achieved with a lower annotation cost than a fully-supervised version by the order of about 10. The smart annotation alone errors were overall lower than those resulting from the segmentation and classification of test data, using the model trained by smart annotation. This is possibly due to the fact that the smart annotation alone results necessarily contained the correct number of steps due to the forced alignment stage. This need for the repetition number is a limitation of this method. However, obtaining the number of cycles in a given section of data is still a lower annotation cost than the full, manual segmentation of the same data section needed for supervised training. Furthermore, the test data were a continuous sequence with mixed classes and transitions between these classes; the training set was small; the test subjects were not included in training; finally, the results were based on the model trained by the smart annotation method not a fully-supervised method.

The classification results and confusion matrix for the smart annotation technique and the model trained in a fully-supervised manner are comparable. The F1-scores of the method using smart annotation and the equivalent fully-supervised version differ by less than 1.0% (absolute). This implies that the smart annotation does not greatly change the accuracy of the model, while it does noticeably change the annotation cost. This also shows that a model can be trained on semi-supervised training data, a subset of the available data, and still be applicable for the annotation of the complete dataset.

The classification results of the model trained with the smart annotation method are not only comparable to the supervised version of the algorithm implemented in this paper, but also correspond to the values found by Panahandeh et al., who used a fully-supervised simultaneous segmentation and classification method [29]. The results are comparable because the datasets are of a similar size and age range, although Panahandeh et al. used a training set of 10 people and a test set of four people. Furthermore, the models and features are similar. Panahandeh et al. found true positive values for rest, walking and running of 99.0%, 97.5% and 98.5%, respectively [29]. The smart annotation method,

requiring a much lower annotation cost, achieved true positive values for rest, walking and running of 99.5%, 97.8% and 99.6%, respectively.

Human error in the reference label would have influenced the error calculations, in the form of inconsistent labeling. An example of this is shown in Figure 6 where the choice of SB is closer to the beginning of the mid-stance phase for Subject A and more towards the end for Subject B. Although this is shown as an error between the result and reference, the resulting segmentation is not unreasonable; in fact, the outcome of the algorithm may be more consistent than the human labelers. The errors were also influenced by the model allowing internal state sequences where some internal states last only one sample and correspondingly have a low probability, as estimated by the model. From our physical understanding of walking, we could force the model to not consider possibilities where the internal states are too short. However, this may limit the model's general applicability; therefore, we did not correct for this in this study. This also shows that the method is robust to noisy annotations.

As illustrated in Figure 6, the internal states could be further investigated. Once calibrated with a gold standard such as that used by [2], they could be used to segment cycles, as well as phases. There is no reliable reference for the gait phases for this dataset; however, from visual inspection, it can be seen that the changes of state often correspond to local maximums or minimums. The most notable is State Change 6–7, which corresponds closely to the heel strike peak according to Rampp et al. [34]. In domain-specific applications, important phases within a cycle could be identified using the internal states.

The differentiation of the classes of walking and running could still be improved using a post-processing method such as what Mannini et al. propose [35]. A possible reason for some of the misclassifications between these two classes is both their similarity and the subjective view of the annotators as to the differentiation of the two classes. This is due to the fact that the dataset contained transitions between running and walking that were not instantaneous. Therefore, misclassifications of less than 1% are to be expected.

One major advantage of this method is that it does not intrinsically rely on a model of the human body, meaning that it could potentially be applied to any cyclic data. It does not rely on a specific shape of a cycle or a specific point of interest at the boundaries; rather, it relies on the data being cyclic. This means that it could potentially be applied to a variety of sensor modalities and positions.

8. Conclusions

The analyses of cyclic movements are restricted by a high annotation cost due to the currently available fully-supervised algorithms for segmentation on a per cycle basis. Several semi-supervised techniques exist that have a lower annotation cost; however, they still require full, manual segmentation of a subsection of the dataset. The available unsupervised methods either require multiple sensors and a model of the body or predictably segment only on the activity level. This paper introduces and demonstrates a method for smart annotation of cyclic data from just one sensor, using an hHMM. Our method had a mean absolute segmentation error of 0.041 ± 0.020 s and a mean cycle duration error of 0.032 ± 0.005 s. When applied to simultaneous segmentation and classification, our method had a similar classification accuracy, 96.2%, to that of the fully-supervised version of the same model, 96.8%, but achieved this with a much lower annotation cost by a factor of about 10. This is a first step towards zero-resource learning of the cyclic data intrinsic to daily life. The application of this method is not limited to cyclic movement data, but could also be applied to other cyclic data in the wearable computing domain, such as other biological signals, as it is not restricted by assumptions based on sensor type or location. This method lowers the annotation cost of training a model capable of continuous monitoring of cycle characteristics such as the progress of a movement disorder or analysis of running technique.

9. Outlook

The presented method still requires the number of cycles per data sample and the class for training; however, these labeling costs are still lower than that of a fully-labeled training set required for state of the art segmentation methods. This need for cycle number and class are the next problems to be tackled, along with automatically splitting the data into the small sections required to train an hHMM. Based on the methods presented in this paper, we will now investigate the automatic detection of the number of cycles present in a section of cyclic data. A clustering method could be investigated for the solving of the class problem for the input data.

The method is based on the concept that hHMM can cope with noisy labels and therefore could also be applied simply in the case of noisy data or a mixed case of some noisy data and some data labeled by cyclic sections. A further application of this model by exploiting the nature of an HMM makes use of the internal states. As seen in Figure 6, the internal states may correspond with a meaningful physical state. Finally, we would like to continue testing our method on cyclic sensor signals common to the wearables field and investigate the effect of dataset size on the accuracy of the smart annotation.

Acknowledgments: This work was in part supported by the FAUEmerging Fields Initiative (EFIMoves). Bjoern Eskofier gratefully acknowledges the support of the German Research Foundation (DFG) within the framework of the Heisenberg professorship program (Grant Number ES 434/8-1). All authors acknowledge support by Deutsche Forschungsgemeinschaft and Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) within the funding programme Open Access Publishing.

Author Contributions: C.F.M., F.H. and B.M.E. conceived of and designed the experiments. C.F.M. performed the experiments and analyzed the data. F.H. and C.S. contributed materials/analysis tools. C.F.M. wrote the paper. All coauthors critically discussed the methods and reviewed the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. This study was partly funded by an unrestricted grant of Bosch Sensortec. Christina Strohmann is employed by Bosch Sensortec. The described method is not used in a commercial product.

Abbreviations

The following abbreviations are used in this manuscript:

hHMM	hierarchical hidden Markov models
SA	smart annotation
IMU	inertial measurement unit
SB	stride border
GMM	Gaussian mixture model

References

1. Klucken, J.; Barth, J.; Kugler, P.; Schlachetzki, J.; Henze, T.; Marxreiter, F.; Kohl, Z.; Steidl, R.; Hornegger, J.; Eskofier, B.; et al. Unbiased and mobile gait analysis detects motor impairment in Parkinson's disease. *PLoS ONE* **2013**, *8*, e56956.
2. Hebenstreit, F.; Leibold, A.; Krinner, S.; Welsch, G.; Lochmann, M.; Eskofier, B.M. Effect of walking speed on gait sub phase durations. *Hum. Mov. Sci.* **2015**, *43*, 118–124.
3. Hughes, N.; Tarassenko, L.; Roberts, S. Markov models for automated ECG interval analysis. *NIPS* **2003**, 611–618.
4. Brajdic, A.; Harle, R. Walk Detection and Step Counting on Unconstrained Smartphones. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'13), Zurich, Switzerland, 8–12 September 2013.
5. Barth, J.; Oberndorfer, C.; Pasluosta, C.; Schüle, S.; Gassner, H.; Reinfelder, S.; Kugler, P.; Schuldhuis, D.; Winkler, J.; Klucken, J.; et al. Stride segmentation during free walk movements using multi-dimensional subsequence dynamic time warping on inertial sensor data. *IEEE Sens. J.* **2015**, *15*, 6419–6440.

6. Mannini, A.; Trojaniello, D.; Croce, U.; Sabatini, A. Hidden Markov Model-Based Strategy for Gait Segmentation using Inertial Sensors: Application to Elderly, Hemiparetic Patients and Huntington's Disease Patients. In Proceedings of the 2015 37th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 5179–5182.
7. Crea, S.; De Rossi, S.M.M.; Donati, M.; Reberšek, P.; Novak, D.; Vitiello, N.; Lenzi, T.; Podobnik, J.; Muni, M.; Carrozza, M.C. Development of gait segmentation methods for wearable foot pressure sensors. In Proceedings of the 2012 International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), San Diego, CA, USA, 28 August–1 September 2012; pp. 5018–5021.
8. Liu, R.; Chen, T.; Huang, L. Research on human activity recognition based on active learning. In Proceedings of the 2010 International Conference on Machine Learning and Cybernetics (ICMLC), Qingdao, China, 11–14 July 2010; Volume 1, pp. 285–290.
9. Miu, T.; Missier, P.; Plotz, T. Bootstrapping Personalised Human Activity Recognition Models Using Online Active Learning. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK, 26–28 October 2015; pp. 1138–1147.
10. Hong, J.; Ramos, J.; Dey, A.K. Toward Personalized Activity Recognition Systems with a Semipopulation Approach. *IEEE Trans. Hum. Mach. Syst.* **2016**, *46*, 101–112.
11. Maekawa, T.; Watanabe, S. Unsupervised Activity Recognition with User's Physical Characteristics Data. In Proceedings of the 2011 15th International Symposium on Wearable Computers (ISWC), San Francisco, CA, USA, 12–15 June 2011; pp. 89–96.
12. Stikic, M.; Larlus, D.; Ebert, S.; Schiele, B. Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2521–2537.
13. Yao, L.; Nie, F.; Sheng, Q.Z.; Gu, T.; Li, X.; Wang, S. Learning from less for better: Semi-supervised activity recognition via shared structure discovery. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16), Heidelberg, Germany, 12–16 September 2016; ACM Press: New York, NY, USA, 2016; pp. 13–24.
14. Zhou, F.; De la Torre, F.; Hodgins, J.K. Hierarchical Aligned Cluster Analysis for Temporal Clustering of Human Motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 582–596.
15. Kruger, B.; Vogele, A.; Willig, T.; Yao, A.; Klein, R.; Weber, A. Efficient Unsupervised Temporal Segmentation of Motion Data. *IEEE Trans. Multimed.* **2017**, *19*, 797–812.
16. Kohlmorgen, J.; Lemm, S. A dynamic HMM for on-line segmentation of sequential data. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, BC, Canada, 3–8 December 2001; MIT Press: Cambridge, MA, USA, 2001.
17. Kulic, D.; Takano, W.; Nakamura, Y. Online Segmentation and Clustering From Continuous Observation of Whole Body Motions. *IEEE Trans. Robot.* **2009**, *25*, 1158–1166.
18. Fox, E.B.; Hughes, M.C.; Sudderth, E.B.; Jordan, M.I. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. *Ann. Appl. Stat.* **2014**, *8*, 1281–1313.
19. Riboni, D.; Szttyler, T.; Civitarese, G.; Stuckenschmidt, H. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16), Heidelberg, Germany, 12–16 September 2016; ACM Press: New York, NY, USA 2016; pp. 1–12.
20. Thomas, O.; Sunehag, P.; Dror, G.; Yun, S.; Kim, S.; Robards, M.; Smola, A.; Green, D.; Saunders, P. Wearable sensor activity analysis using semi-Markov models with a grammar. *Pervasive Mob. Comput.* **2010**, *6*, 342–350.
21. Joukov, V.; Bonnet, V.; Karg, M.; Venture, G.; Kulic, D. Rhythmic Extended Kalman Filter for Gait Rehabilitation Motion Estimation and Segmentation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, doi:10.1109/TNSRE.2017.2659730 .
22. Aoki, T.; Venture, G.; Kulic, D. Segmentation of Human Body Movement Using Inertial Measurement Unit. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 1181–1186.
23. Mannini, A.; Trojaniello, D.; Cereatti, A.; Sabatini, A. A Machine Learning Framework for Gait Classification Using Inertial Sensors: Application to Elderly, Post-Stroke and Huntington's Disease Patients. *Sensors* **2016**, *16*, 134.

24. Andreao, R.V.; Dorizzi, B.; Boudy, J. ECG signal analysis through hidden markov models. *IEEE Trans. Biomed. Eng.* **2006**, *53*, 1541–1549.
25. Fine, S.; Singer, Y.; Tishby, N. The Hierarchical Hidden Markov Model: Analysis and Applications. *Mach. Learn.* **1998**, *32*, 41–62.
26. Lv, F.; Nevatia, R. *Recognition and Segmentation of 3-D Human Action Using HMM and Multi-Class AdaBoost*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 359–372.
27. Lin, J.F.S.; Kulic, D. Online Segmentation of Human Motion for Automated Rehabilitation Exercise Analysis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2014**, *22*, 168–180.
28. Rabiner, L. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286.
29. Panahandeh, G.; Mohammadiha, N.; Leijon, A.; Handel, P. Continuous Hidden Markov Model for Pedestrian Activity Classification and Gait Analysis. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 1073–1083.
30. Juang, B.H.; Rabiner, L. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. Acoust. Speech Signal Process.* **1990**, *38*, 1639–1641.
31. Brugnara, F.; Falavigna, D.; Omologo, M. Automatic segmentation and labeling of speech based on Hidden Markov Models. *Speech Commun.* **1993**, *12*, 357–370.
32. Steidl, S.; Riedhammer, K.; Bocklet, T. Java visual speech components for rapid application development of GUI based speech processing applications. In Proceedings of the 12th Conference of the International Speech Communication Association (INTERSPEECH 2011), Florence, Italy, 28–31 August 2011.
33. Taborri, J.; Palermo, E.; Rossi, S.; Cappa, P. Gait Partitioning Methods: A Systematic Review. *Sensors* **2016**, *16*, 66.
34. Rampp, A.; Barth, J.; Schüle, S.; Gaßmann, K.; Klucken, J.; Eskofier, B. Inertial Sensor Based Stride Parameter Calculation from Gait Sequences in Geriatric Patients. *IEEE Trans. Biomed. Eng.* **2014**, *62*, 1089–1097.
35. Mannini, A.; Sabatini, A.M. Gait phase detection and discrimination between walking-jogging activities using hidden Markov models applied to foot motion data from a gyroscope. *Gait Posture* **2012**, *36*, 657–661.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).