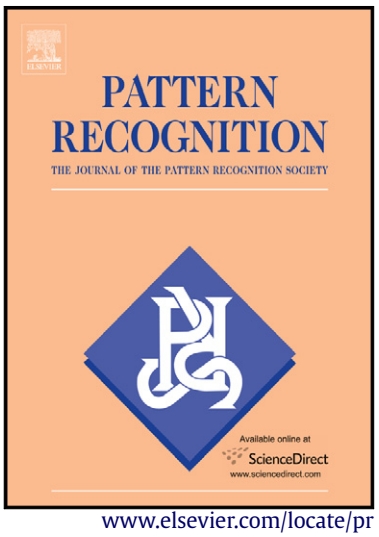


Trajectory-based Human Action Segmentation

Luís Santos , Kamrad Khoshhal , Jorge Dias

PII: S0031-3203(14)00329-X
DOI: http://dx.doi.org/10.1016/j.patcog.2014.08.015
Reference: PR5198

To appear in: *Pattern Recognition*

Received date: 26 June 2013
Revised date: 7 July 2014
Accepted date: 17 August 2014

Cite this article as: Luís Santos , Kamrad Khoshhal , Jorge Dias , Trajectory-based Human Action Segmentation, *Pattern Recognition,* http://dx.doi.org/10.1016/j.patcog.2014.08.015

Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 00 (2014) 1–27

Pattern Recogni-tion

Trajectory-based Human Action Segmentation

Luís Santos[a], Kamrad Khoshhal[a], Jorge Dias[a,b]

[a]*Institute of Systems and Robotics (ISR), University of Coimbra, Portugal*
[b]*Khalifa University of Science, Technology & Research (KUSTAR), Abu Dhabi, UAE*

Abstract

This paper proposes a sliding window approach, whose length and time shift are dynamically adaptable in order to improve model confidence, speed and segmentation accuracy in human action sequences. Activity recognition is the process of inferring an action class from a set of observations acquired by sensors. We address the temporal segmentation problem of body part trajectories in Cartesian Space, in which features are generated using Discrete Fast Fourier Transform (DFFT) and Power Spectrum (PS). We pose this as an entropy minimization problem. Using entropy from the classifier output as a feedback parameter, we continuously adjust the two key parameters in a sliding window approach, to maximize the model confidence at every step. The proposed classifier is a Dynamic Bayesian Network (DBN) model where classes are estimated using Bayesian inference. We compare our approach with our previously developed fixed window method. Experiments show that our method accurately recognizes and segments activities, with improved model confidence and faster convergence times, exhibiting anticipatory capabilities. Our work demonstrates that entropy feedback mitigates variability problems, and our method is applicable in research areas where action segmentation and classification is used. A working demo source code is provided online for academical dissemination purposes.

Keywords: Motion Segmentation, Classification Framework, Signal Processing, Motion Variability, Adaptive Sliding Window.

1. Introduction

Action recognition is an active research topic within the scientific community, with several applications, which include human-machine interfaces, intelligent video surveillance, video indexing and analysis, to name just a few. The action segmentation problem is a key issue in action recognition and may be divided into two stages: (1) Learning and (2) Classification. The learning stage often involves a data preprocessing step to find alternative, discriminant representations for different properties of the input signal. In this work, we consider a data driven probabilistic representation for the action model, which is learned from a set of training data. This action model is posteriorly used to identify to which action class each observable feature belongs.

Email addresses: luis@isr.uc.pt (Luís Santos), kamrad@isr.uc.pt (Kamrad Khoshhal)

A popular applied method to this problem is the sliding window approach. The window is used to progress sequentially through the input signal, creating data segments from which features are extracted. This method is popular because of its direct integration with the majority of classification algorithms. However, fixed parameter values are a significant cause of classifier under-performance: slow convergence and/or borderline decisions (e.g.[1]). Choosing the ideal parameter values is not a trivial task and an optimal selection may differ for different performers and/or actions. Thus in this paper, we present a dynamically adaptive sliding window, where classification entropy is used to adjust the window length and time shift parameters at every step.

1.1. Action Segmentation Issues

The execution of actions differs from person to person. Factors like rigidly defined performance instructions, mobility restrictions introduced by the experimental set-up, cultural or anatomical characteristics are known to introduce variability. The majority of action models usually rely on a set of assumptions, which interfere with the classification of live executions and present some challenges. In our work, we are addressing the following problems:

- Frameworks can present high classification accuracy and the majority of the correct decisions are of low confidence. This is specially true as the number of different actions grows.
- The time it takes for a model to make a decision is highly dependent on the generated features, whereas being able to anticipate a decision is an issue of interest for an accurate temporal segmentation.

Approaches within action segmentation somehow try to address these factors. In this research, we are focused on extending our previous work using a fixed length sliding window approach [2, 3], improving our segmentation solution to cope with classification performance issues. A survey on action segmentation [4] identifies other works which also use fixed length sliding windows [5, 6, 7, 8]. In some of these works, the classification framework is augmented with multiple concurrent classifiers using windows of different lengths at the expense of increasing computational cost. Supported by examples in literature, the following paragraphs summarize the main key problems in fixed parameter sliding window approaches.

A sliding window approach with fixed parameters is used in [9] to detect events in long video sequences. They analysed the delay (measured in frames) between ground truth annotations and the output of a classifier using the following parameters: a window size of 64 frames and a 8 frame time shift. Since an event temporal duration is variable, the fixed sliding window caused sample misclassification. In [10], the size of the sliding window is given in seconds (4 seconds) and it was used to detect unusual activities in video sequences. Result analysis shows that segmentation is not perfect and the reason for such large window size was to make sure that the buffer had enough signal information. Consequently, these large data samples contained higher rates of outlier information, which

increases the number of borderline decisions. In [11] a sliding window was tested with two different sized, 48 and 24 frames. These were applied to video segmentation in the classification of human actions. Experimental results were presented without including classification decisions which contain transition from one action to another. Despite the application of this strategy, excluding transition frames did not prevent segment misclassification.

In other works, sliding window approaches are integrated with other techniques. For example, they can be integrated with Dynamic Time Warping [12, 13], or Grammars [14, 15]. However, methods that allow to dynamically adjusting the sliding window parameters in action segmentation are rarely explored. In [16], the window parameters are adjustable from sensor based events and dependent on the signal processing techniques. However, authors conclude that their approach is restricted by the application of the selected algorithms and sensors. In [17], a new type of self-adaptive sliding window is proposed for data mining. The parameters are adjustable based on the signal properties. While results show to be satisfactory, the success of the proposed technique depends on the existence of specific signal properties. We were not able to find in the literature sliding approaches with dynamic parameters that are independent of the type of signal properties or processing algorithms.

1.2. Other Works Related on Action Segmentation

A recent survey by Weinland et al. [4], has identified three major action segmentation categories: Sliding Window , Boundary detection and Grammar Concatenation. The already reviewed **Sliding windows** are used to divide a motion sequence into multiple overlapping segments, which are bounded by the window limits. The information within the window may or may not be processed for alternative representations. Each candidate segment (or equivalent representation) is then used for sequential classification. The success of this approach strongly depends on the discriminant abilities of the generated representations. As mentioned this technique is easily integrated with the majority of static and dynamic classifiers. The major drawbacks of this technique are computational burden, and the need of multiple window sizes to overcome the variability problem. **Boundary detection** methods generally identify discontinuities or local extrema in observed motion signals. The boundaries usually define an implicit basic action taxonomy, without however depending on specific class definitions. A branch of works identify boundary at the cost of the dynamics of the observed signal, such as [18, 19]. Others depend on geometric property changes observed through techniques like Principal Component Analysis [20] or piecewise arc fitting models [21, 22]. A related research addresses the segmentation problem from the subspace separation perspective, exploring the so called Agglomerative Lossy Compression [23]. In [24], the authors apply Singular Value Decomposition (SVD) to a long sequence of optical flow images in order to detect trajectories discontinuities within SVD component trajectories. Ogale et al. [25] also explore optical flow of body silhouettes, performing segmentation by detecting minima and maxima values of the absolute value sequence. A method using features from visual hulls is developed in [26]. This category of approaches is very sen-

sitive to noise and other related errors (e.g. camera perspectives). Additionally, it allows generic segmentation, but is not particularly suitable for labelling purposes. The focus is on boundary identification rather than interpretation of intermediate data. Lastly, Weinland et al. [4] identify **Grammars** as another category. The common approach is to model state transitions between actions, where Hidden Markov Models (HMM) are a popular approach. Multiple methods can be used to generate features. Some examples are curvature scale space and centroid distance function [27], joint angles alone [28, 29], or together with velocity profiles [30], dynamic system representations [31, 32, 33] and geometrical property encoding [34]. These are applied to segment and label action sequences, at the expense of computing a minimum-cost path through the model using techniques like Viterbi path, Conditional Random Fields or Markov Models. However, these methods rely on the comprehensiveness of state grammars, which may jeopardize the model effectiveness and the generalization purpose, if large amount of training data is not available.

We can say that temporal action segmentation is implicitly addressed in most problems of action classification at some point of their research. The majority of research is done in computer vision and applied to image sequences, where each frame is classified consequently generating a temporal sequence of associated action labels, such as in [35, 36]. More classical vision-based approaches only consider data from the current image frame, attempting to find a class that represents the acquired data more closely. There are in fact other works that consider collections of multiple images, as it happens in a sliding window paradigm. But again, these also use a pre-defined number of images and time shifts (e.g. [37]).

1.3. Definitions and Problem Statement

A motion instance is defined as a contiguous sequence of human body movements, which is composed of a concatenation of different actions. Let motion instance Ω be a sequence of 3-D Cartesian coordinates Y, defining a discrete trajectory of random duration T (measured in *frames*), for a body part such that:

$$\Omega = \begin{bmatrix} Y_1 \\ \vdots \\ Y_T \end{bmatrix}, Y \in \mathbb{R}^3 \text{ and } T \in \mathbb{N} \tag{1}$$

In the processing stage, Ω is divided into multiple, **overlapping** segments δ, generated upon using a sliding window of length ω_t *frames* and each δ is separated in time by a time shift Δ_t, such that:

$$\delta_t \subset \Omega : \delta_t = \begin{bmatrix} Y_{t-\omega_t} \\ \vdots \\ Y_t \end{bmatrix}, \omega_t < T \tag{2}$$

At this point, let us introduce the following two key definitions in sliding window approaches:

Definition 1 ω_t **(Window Length).** *Also known as window size, it corresponds to the number of contiguous sensor readings (in our work, Cartesian Coordinates Y) that are contained within the window, i.e. how much of the captured trajectory is used to generate a segment* δ*. The length* ω_t *is implicitly defined in equation (2) and is measured in frames*;

Definition 2 Δ_t **(Time Shift).** *Corresponds to the displacement between two consecutive windows, measured in frames, which is equal to the difference between the index of the first frame in each window. More specifically, let it be a* $\delta_1 = \left[Y^1_{t-\omega_t}, \cdots, Y^1_t\right]^{tr}$ *and* $\delta_2 = \left[Y^2_{t'-\omega_t}, \cdots, Y^2_{t'}\right]^{tr}$ *such that the time shift* $\Delta_t = t' - t$*; The subscript* ***tr*** *represents the transpose of a matrix. Please note that the time shift can be defined in either frames or seconds, where the time shift in seconds is given by the ratio between the time shift in frames and the acquisition frequency , i.e.* $\Delta_t[seconds] = \frac{\Delta_t[frames]}{f[Hertz]}$.

To avoid using the raw segment data, each δ_t is transformed into a representative feature vector $\mathcal{V}$, of lower dimension, for which a transformation function exists, such that $\delta \mapsto \mathcal{V} : \{v_1, \cdots, v_i\} \in \mathcal{V} = g(\delta)$. Our framework uses two different class layers for analysing motion instances. One corresponds to a set C of motion descriptors defined upon Laban Movement Analysis (LMA) [38], where $c_n \in C$ is a variable representing the n^{th} Laban component. These components are defined and used in LMA to characterize human motion in its different geometrical, kinematic and expressive properties. The other layer emerges as a combination of variables c_n, and defines the action space $\Lambda = \{\beta_1, \cdots, \beta_a\}$ Consider a movement sequence which is a concatenation of N action segments β, where each β is a **non-overlapping** sub-set of Ω. A single state of each $c_i, i = 1, \cdots, n$ is assigned to each segment β during a supervised learning approach. The challenge is devising an association process to learn the action model, envisioning its separability capabilities. The model is posteriorly used in a classification process, from which the temporal segmentation of Ω is derived.

$$\beta_j = \begin{bmatrix} Y \\ \vdots \end{bmatrix} \overset{c_n}{\longleftarrow} C \, , \beta_j \in \Omega \tag{3}$$

Consider a new action β, for which applying a sliding window approach generates multiple segments δ. Most misclassified samples have their errors emerging from the incorrect selection of the fixed window parameters. Therefore, we hypothesize that adapting these parameters at each step will improve classification, thus coping with the variability of different performances of the same action. In fact, rather than selecting a method to optimize the fixed window parameters, our main challenge is to formulate a model, which iteratively readjusts the length and the time shift based on entropy feedback and knowledge of previous parameter definitions. Table 1 summarizes the relevant variables,

which are used throughout this article.

Problem - Given an activity sequence Ω, find the current window length ω_t that best fits the current segment δ and minimizes the classification entropy h over the variables $c_n \in C$.

$$\omega_{t+1} = g(h_t, \omega_t) \Rightarrow min(h_t) \tag{4}$$

Additionally, when uncertainty is high (e.g. on class transition), adjust the time step so the classifier can adapt to changes without diverging to misclassified samples.

$$\Delta_{t+1} = g(h_t, \omega_t, \Delta_t) \Rightarrow min(h_t) \text{ and } \downarrow errors \tag{5}$$

Consider sequences to be subject to noise and instance variability for the same actions performed at different instants of time.

$$\Omega' = \Omega + \eta \tag{6}$$

where η is a source of additive white noise.

1.4. Our Approach

In our work, we are addressing temporal action segmentation of body part trajectories generated upon random human activity performances, as an extended solution to our fixed sliding window classifiers in action recognition [2, 3]. To acquire 3-D trajectories from different body parts, we are using a Motion capture (Mo-Cap) device, which is synchronized with a video sequence I of activity performances. Feature vectors are computed upon application of a Discrete Fast Fourier Transform (DFFT) to the acceleration signals generated from the acquired body part trajectories. This feature approach has been previously applied with success in human motion analysis problems [39]. To learn the action model, we apply a mixture model based approach, a popular methodology in action segmentation and recognition, for which we have past experience [2, 3]. The sliding window approach requires the learning process to

Table 1: Summary of relevant variables.

Variable	Set	Space
v	$\mathcal{V}$	Low-level Features
c_n	C	Laban Descriptors
β	Λ	Action
ω_t (or ω)		Window Length
Δ_t (or Δ)		Time Shift
h		Classification Entropy

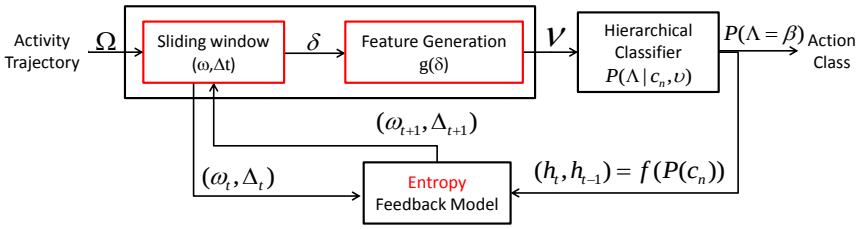


Figure 1: Simplified Block Diagram providing an overview of the proposed approach.

be supervised, as it plays a crucial role for the success or failure of the model [4]. The learned conditional models are integrated in a Dynamic Bayesian Network classifier, which applies Bayesian inference and is used to segment an activity sequence using a *maximum a posteriori* (MAP) approach.

In our experimental set-up, two different parameters are adapted, both independently and simultaneously. One strategy adapts the window length ω_t and is referred to, using the acronyms ***Adapt***-ω $(\omega_{min}, \omega_{max})$ or ***Fix***-ω, considering whether we are using the adaptive or fixed approach respectively. The other is concerning the time shift Δ_t. The acronyms for this approach are ***Adapt***-Δ or ***Fix***-Δ for adaptive and fixed strategies. Acronyms are then combined, so to allow identifying the applied strategies. Our proposed adaptive sliding window methodology (illustrated in Figure 1), is presented as an improvement to classic fixed sliding window classification methods which:

- shows increased classification confidence;
- increases the classifier speed therefore anticipating the decision;
- dynamically adapts to different sources of performance variability.

Figure 2 encompasses the proposed concept illustration, of the adaptive parameter based on entropy feedback and knowledge of previous parameters.

1.5. Paper Structure

We first introduce the feature generation within the fixed parameter sliding window paradigm (Section 2.1), showing how different parameter values affect the learning distributions in Section 2.2, testing separability criteria and other relevant metrics. The classification framework is presented in Section 3.2, where our proposed method for adapting the sliding window parameters is explained in Sections 3.3 and 3.4. The action segmentation experiments are present in Section 3.5, where the experimental set-up is explored using both fixed and adaptive parameter approaches. We complement our research with a discussion of how our approach allows to anticipate classification decisions on Section 3.5.1. This work concludes with a discussion over experimental results (Section 4), future work and the expected impact in related research area.

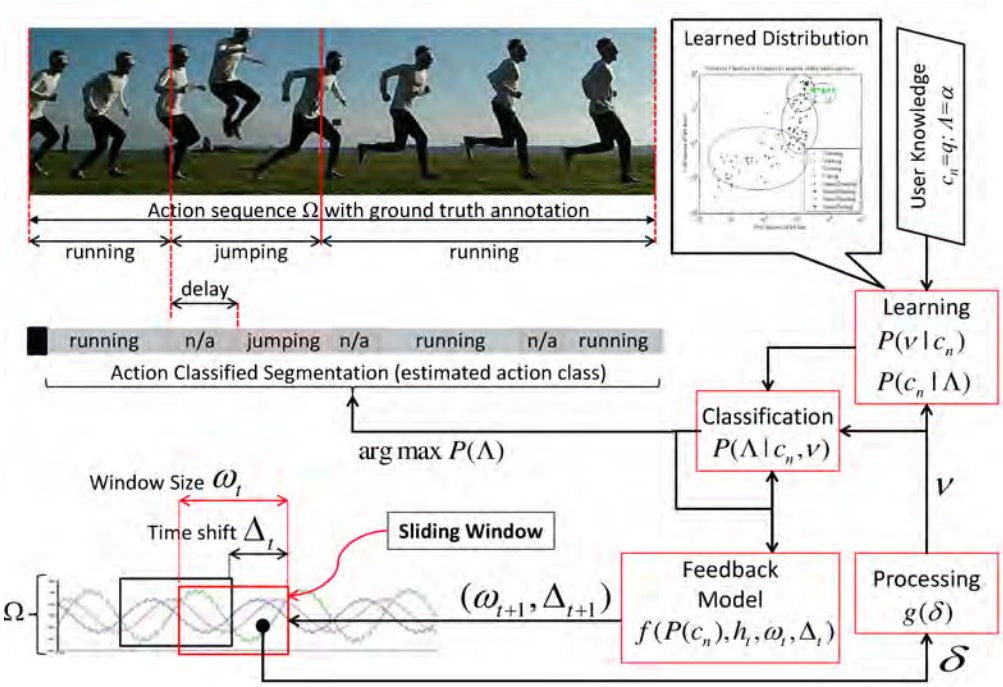


Figure 2: Scheme of the proposed concept along with the block diagram which formally describes our framework. An activity is segmented using a sliding window, whose parameters are adaptive based on entropy feedback. We learn body Laban and Action model, which are manually annotated within a supervised learning approach. To segment an activity in different actions, we select the most probably action Λ from our hierarchical classifier.

2. Learning the Action Model

In this section, the trajectory feature generation process is presented and also how different window size values influence the resulting probability distributions, upon application of the learning strategy.

2.1. Preprocessing

Our work emerges as an improved classification strategy to our previously developed research in action recognition, where features are represented in the frequency domain. An acceleration time series is computed from the Cartesian trajectories. Then, the Discrete Fast Fourier Transform (DFFT) and signal Power Spectrum (PS) are applied. Let the segment δ be bounded by a sliding window of length l, such that:

$$\delta = \begin{bmatrix} Y_1 \\ \vdots \\ Y_l \end{bmatrix}, Y \in \mathbb{R}^3 \tag{7}$$

Given the segment trajectory δ we compute acceleration $a_t = \frac{\Delta v}{\Delta t}$, where $v_t = \frac{\Delta Y}{\Delta t}$. The generated acceleration sequence $a(t) = a_1, ..., a_t$ will be decomposed using DFFT algorithm, generating the list of coefficients x of a finite combination

of complex sinusoids, ordered by their frequency.

$$a(t) = \sum_{n=0}^{l-1} x_n e^{\kappa}, \text{ with } \kappa = \frac{-i2\pi kn}{l} \tag{8}$$

We can then calculate the PS of the acceleration signal, knowing that $a(t)$ is a finite energy signal, as:

$$\Phi(\omega) = \left| \frac{1}{\sqrt{2\pi}} \sum_{\infty}^{-\infty} a(t) e^{i\omega t} \right|^2 \tag{9}$$

The continuous approach can be generalized to discrete, for which we are able to compute the energy spectral density.

Feature variables are generated upon dividing the PS coefficient value ranges into four distinct classes as depicted in (10).

$$\mathcal{V} = \{no, low, medium, high\} \tag{10}$$

Further details on the presented feature generation process can be found in [2, 38].

2.2. Learning

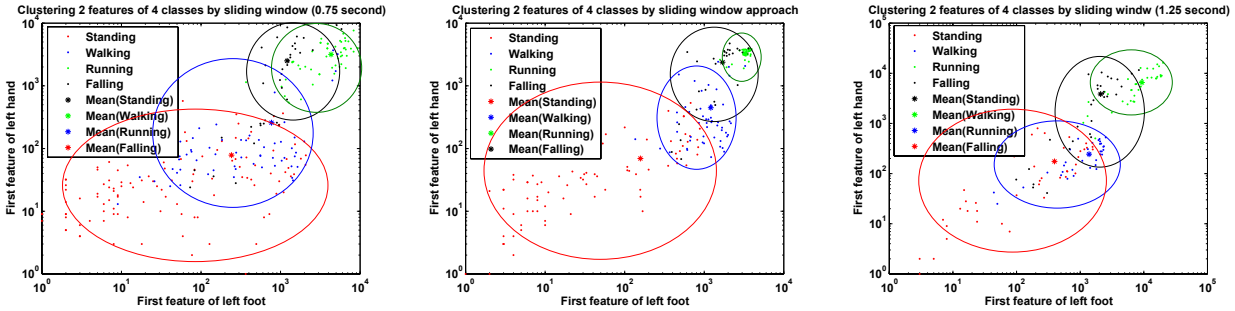
The learning method follows a Mixture Model approach, in which feature vectors are clustered according to a class of c_n they belong, for example, grouping all segments labelled with $c_1 = sudden$. This process is done through supervised learning methodology (which has been conducted *offline*). The mixture obeys the following Gaussian decomposition:

$$P(\mathcal{V}|C) = \sum_{i=1}^{n} \phi_i \, g(c_i|\mu_i, \sigma_i) \tag{11}$$

where class c_i is represented by an average vector μ_i and a covariance matrix σ_i. To **evaluate the action model**, we assess class variance (an indicator of dispersion,) and a separability criteria for measuring inter-class distances. Variance σ_i is estimated directly from the solution of the Mixture Model formulated in equation (11), using an Expectation Maximization approach. To measure the separability between two classes, a popular measure is the Fisher's Discriminant (FD) [40]. Rao [41] generalized the FD to more than two classes, using an extended formulation to find the subspace containing all class variability. First we define the class scatter as:

$$S_c = \frac{1}{n_i} \sum_{j=1}^{n} (x_j - \mu_i)(x_j - \mu_i)^T \tag{12}$$

where n_i is the number of samples for a given class c_i, while μ_i represents the mean of that same class c_i. From the class scatter, we can compute the *within class* scatter $S_W = {}^{n_i}/_{n} \sum_{j=1}^{c} S_j$, with n the total number of samples.



(a) $\omega = 90$ $frames$ (0.75 s) (b) $\omega = 120$ $frames$ (1.00 s) (c) $\omega = 150$ $frames$(1.25 s)

Figure 3: Class clusters for 4 different actions: standing, walking, running and falling. The images represent clusters computed upon a fixed window approach of: 90 ; 120 and 150 frames (sampled at 120Hz). The points correspond to the features generated from the trajectories captured from the left foot of the acting person, during the different trials of each action.

Considering the Gaussian Mixture Model defined in (11), the *between class* variability can be defined for each class as:

$$S_B = \sum_{i=1}^{c} \frac{n_i}{n}(\mu_i - \mu)(\mu_i - \mu)^T \tag{13}$$

where μ is the mean of class means μ_c. The class separability will be given by

$$J = \frac{det(V^T S_B V)}{det(V^T S_W V)} \tag{14}$$

Vector V is computed by solving the eigenvalue problem $S_B V = \lambda S_W V$, where V is the eigenvector corresponding to the largest eigenvalue.

2.3. Experimental Learning Results

We now demonstrate how different lengths have direct impact in the supervised learning process. The presented results aim to show that selected values for length ω_t have consequences which are reflected in the action model, both visually and through adequate metrics. This impact of parameter selection naturally propagates to the classification algorithm (as is demonstrated in several works such as [9] or [10]), and thus, in the entropy. The class clusters for the 2 dimensions of the feature vector are presented in Figure 3 using three different fixed window sizes. When using $\omega = 0.75$ *seconds*, we observe an overlap between class pairs *standing-walking* and *running-falling*. In the 1 *second* case, class *running* is completely inside *falling*, whereas with $\omega = 1.25$ *seconds*, there are multiple overlapping regions. Let us recall that the DFFT is being applied to the acceleration signal, therefore *falling* and *running* fall in the high acceleration signals while *standing* is mostly a static activity and *walking* is situated in between. Most importantly, we can visually verify that changes in the length of the window size are reflected in the class learning

process. We extend our analysis using the quantitative metrics presented in Table 2. Variables S_i represent scatter measures for each class. S_B refers to interclass average covariance, which can be interpreted as a dispersion measure, since it reflects the weighted distance from the class centres to their average value. From the generalized Fisher's

Table 2: Generalized Linear Discriminant Analysis coefficients for the cases presented in Figure 3, where $s_i, i \in \{x, y\}$ are the interclass average variance along each axis.

		Window size (ω)		
		0.75 s	1.00 s	1.25 s
Scatter Val.	S_1 [s_x, s_y]	[1187,2657]	[904,1493]	[1859,3632]
	S_2 [s_x, s_y]	[2273,1671]	[533,572]	[5731,4138]
	S_3 [s_x, s_y]	[443,114]	[245,93]	[465,202]
	S_4 [s_x, s_y]	[1185,643]	[609,548]	[794,148]
	J	7.191	14.279	29.594

discriminant definition, we know that the higher the value of J, the better defined and separated are the learned class distributions. The analysis of Table 2 visibly shows that small changes on ω_t have high impact on class dispersion. The number of points do affect the calculation of the value of S and thus the factor J, however this impact is mitigated as the number of points increases. In fact, the parameters of the Gaussian distributions will tend to converge as the number of samples increases. Thus, the impact of each new point will be $1/n_i$ (as equation 12 intuitively demonstrates), where n_i is the total number of points belonging to a given distribution.

3. Action Classification

Our framework aims to segment actions in different abstraction symbolic levels, by means of a Bayesian classifier. Those levels are:

- Laban Movement Analysis: a set of activity invariant descriptors based on the LMA's components. For example, LMA's component *Effort Time* has two states, *sudden* and *sustained*, while component *Shape* is associated to states such as *reaching* or *retreating*.
- Action: a variable whose states represent different movements as a combination of Laban variables. These correspond to actions like *walking* or *sitting*.

3.1. Experimental Set-up

As already mentioned, the input signal in our experiments is a contiguous sequence of 3-D Cartesian coordinates, acquired at a fixed sampling rate of $f = 120Hz$. These are acquired for different body parts and stationary object positions, generating three-dimensional trajectories and object relative poses. Body part data is acquired using the Inertial Measuring Units of the XSENS Moven Suit (`http://www.xsens.com/products/xsens-mvn/`),

whereas object pose information is retrieved using magnetic sensors from the Polhemus Liberty magnetic tracker (`http://polhemus.com/motion-tracking/all-trackers/liberty/`). The experimental sessions contain long sequences of human body movement, which are composed of different actions: *standing, walking, falling, sitting, rising* and *no move* (absence of movement). The total number of different action performances contained within all sequences sum up to around 100 segments, executed by different persons in a set-up that has been previously used in [2] (For reference, please check `http://mrl.isr.uc.pt/experimentaldata/public/uc-3d`). Actors perform their movements naturally, whereas the only restriction is that they had to perform a certain sequence of different actions, e.g. an actor will get up (*rising*), then will start *running* and will *fall* on the floor at the end of the sequence. The generated spatial and frequency-based features are used as evidence in the Bayesian Classifier Model towards action segmentation.

The results obtained from this research are compared with our previous work mainly because we have privileged access to the data at every step of the process. More concretely we are interested in the frame by frame decisions, which allow an accurate measurement about the temporal improvement of our approach and also the classification confidence measured through the entropy value. However, as it will be demonstrated in the following sections, our approach is not restricted to this experimental set-up. In fact, sliding window approaches are known to be applied to different types of data. The information that we use to adapt the window parameters is completely independent from the type of input signal, making this approach applicable to a wide variety of scenarios.

3.2. Action Classification Model

The action model is a hierarchical framework, in which inference occurs sequentially. To learn the model two strategies are assumed. To associate Laban variables to the frequency based features, we use Gaussian distributions. While learning the action model, a statistical approach is applied, where occurrences of c_n are accounted for and normalized, generating histogram probabilistic distributions. The first layer of the action model is parametrized as:

$$P(laban|feature) = P(laban)\frac{\prod_{q=1}^{i} P(feature_q|laban)}{\prod_{q=1}^{i} P(feature_q)} \tag{15}$$

We will be focusing our attention at this level, because it is where the window parameters will have most of the impact. In fact, the Laban model is learned based on the data bounded by the window. The entropy used to get feedback from the window's parameters is computed from the output $P(laban|feature)$. The action variable states are inferred as a combination of previously estimated *laban* variables. An action is inferred based on:

$$P(action|laban) = P(action)\frac{\prod_{q=1}^{n} P(laban_q|action)}{\prod_{q=1}^{n} P(laban_q)} \tag{16}$$

The estimation occurs using Bayesian inference algorithms, where a *Maximum A Posteriori* (MAP) approach is applied, which is done using numerical approach, given that our formulation poses a closed-form solution. The most probable state for a variable θ upon knowledge from observations x is given by:

$$\hat{\Theta}(x) = arg_{\Theta} max P(\theta) P(x|\theta) \tag{17}$$

The variable states for each abstraction level which present the maximum probability value, are selected as the ones describing the corresponding segment δ, thus segmenting a sequence Ω, as illustrated in Figure 2.

3.3. Adaptive Sliding Window

The classification inference algorithms usually apply fixed parameter sliding windows. However, selecting optimal parameters is not easy. In fact, what can be a good parameter selection for a sequence, might fail to show correct segmentation when using a different performer. Contrary to this classic sliding window approaches, we propose a method which continuously adapts the window parameters. Let us assume the following definitions:

- h = Entropy value.
- H = Entropy time series.
- ω = Window size.
- ω_d = Default window size.
- W = Window size time series.

Consider that for a distribution $p = \{x_1, \cdots, x_n\}$, the maximum value for h is given by $max(h) = log(n)$. Bear in mind, entropy is a normalized value, upon the max(h), such that $h \in [0, 1]$.

3.3.1. Window Size

The size of the sliding window is adjusted upon the following parameters: previous window lengths and the classification entropy. The trends for each of these parameters are also analysed. More specifically, we analyse whether the window size has previously increased or decreased (which is here referred as *scale direction*). The same pattern is checked for the classification uncertainty (given by the entropy value). A numerical representation about the trend of each of these parameters is given by the first and second order backward differences. Using entropy as an example, if $h_{t-1} > h_t$, then the first order backward difference is negative, meaning that the entropy value is decreasing and that our classifier decisions are becoming more *accurate*. By combining the implicit information about these

Table 3: Summary of implicit signal rules. N/R = Not relevant.

dH	h	dW	ω	d^2H	h	$\hat{\omega}$
+	Worst	+	Increasing	N/R	N/R	(-) Shrink
0	Stable	+	Increasing	+	Increasing Tendency	(-) Smaller Shrinkage
0	Stable	+	Increasing	-	Decreasing Tendency	(+) Smaller Growth
-	Good	+	Increasing	N/R	N/R	(+) Growth
+	Worst	-	Decreasing	N/R	N/R	(+) Growth
0	Stable	-	Decreasing	+	Increasing Tendency	(+) Smaller Growth
0	Stable	-	Decreasing	-	Decreasing Tendency	(-) Smaller Shrinkage
-	Good	-	Decreasing	N/R	N/R	(-) Shrinkage

parameters, we establish a set of rules which are used to determine the new window size. The rationale behind our approach is summarized in Table 3.

Let us further reinstate our approach. Assume now the case where the entropy value $h_{t-1} < h_t$, which means that our previous decision lead the model to become *more* uncertain. We analyse this phenomenon in light of the immediate past window sizes ω_{t-1}. Whichever has been our previous decision of increasing or decreasing the window size, it has led to a decreasing model confidence, therefore the window size needs to be corrected in the opposite direction. In the cases where our decision has led to an increase of model certainty, we define that the last decision about the window length is correct and should be maintained.

There are however cases where consecutive instants have equal values for h, i.e. $h_{t-1} = h_t$, for which the backward difference is zero. When such event occurs, we replace the first order backward difference by its second order counterpart, which represents the growth tendency. Equivalent to analysing the second derivative for a continuous time series, we assume that upwards concavity represents tendency to increase and vice-versa. Bear in mind that by analysing a tendency, the scaling factor needs to be constrained when compared to using the first order difference.

3.3.2. Formulation

In light of the presented *rationale*, the basic definition for the window length obeys the following equation:

$$\omega_t = (1 + \alpha)\omega_{t-1} \tag{18}$$

where ω_t is the window length at instant t, and the variable $\alpha = [\alpha_{min}, \alpha_{max}]$ a scaling factor such that:

$$\underbrace{(1 + \alpha_{min})\omega_d}_{\omega_{min}} \leq \omega_t \leq \underbrace{(1 + \alpha_{max})\omega_d}_{\omega_{max}} \tag{19}$$

The **scaling direction** $\vec{\alpha}$ according to the aforementioned rationale, is formulated mathematically as:

$$-\frac{dH}{dt}\frac{dW}{dt} \tag{20}$$

For the special cases where $\frac{dH}{dt} = 0$, this argument is replaced by the second order backward difference $\frac{d^2H}{dt^2}$.

$$-\frac{d^2H}{dt^2}\frac{dW}{dt} \tag{21}$$

However, when $\frac{dH}{dt} = 0$, the second order difference is considered a weak indicator. Therefore, we propose two constraints a and b, such that $\frac{dH}{dt} \geq \frac{d^2H}{dt^2}$. From equations 20 and 21, we obtain:

$$-\frac{dW}{dt}\left(a\frac{dH}{dt} + b\frac{d^2H}{dt^2}\right) \tag{22}$$

We must also consider the specific case where $\frac{dW}{dt} = 0$, which leads to $\vec{\alpha} = 0$. Our solution is making this factor to converge to the default window size, for which equation 22 is rewritten as:

$$(\omega_d - \omega)\left|a\frac{dH}{dt} + b\frac{d^2H}{dt^2}\right| \tag{23}$$

where the derivatives no longer control the scaling direction. In these cases, the direction is controlled by the difference between the current window size and the selected default value. The scaling direction $\vec{\alpha}$ can then be summarized as:

$$\vec{\alpha} = \begin{cases} -\frac{dW}{dt}\left(a\frac{dH}{dt} + b\frac{d^2H}{dt^2}\right) & , \frac{dW}{dt} \neq 0 \\ (\omega_d - \omega)\left|a\frac{dH}{dt} + b\frac{d^2H}{dt^2}\right| & , \frac{dW}{dt} = 0 \end{cases} \tag{24}$$

This latter formulation addresses only the scaling direction. The issue of *how much* (**scale**) should the window grow or shrink is addressed in the following paragraphs. The goal is to obtain a normalized factor that can be put as a percentage value of the previous window size. This factor should be proportional to the margins between the current and maximum/minimum values for window size. In addition, the selected function should be symmetric to the origin, meaning that the sigma of α is defined upon $\vec{\alpha}$. The function in equation (25) encompasses both of these properties.

$$\alpha = \frac{1}{k}\frac{\sqrt{(1 + 4\vec{\alpha}^2)} - 1}{2\vec{\alpha}} \tag{25}$$

where k is an inverse proportional factor which may limit growth (default $k = 1$). Figure 4 illustrates equation (25) for a clearer visualization. One should note that the window size must not scale beyond the limits defined in equation

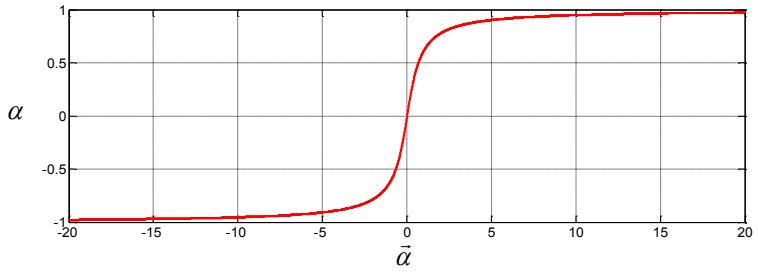


Figure 4: Envelope function for the growth percentage. When $\alpha \to \infty$ then $\vec{\alpha} \to 100\%$

(19). Hence, the following formulation is proposed:

$$\omega_t = \begin{cases} \omega_{t-1} + \alpha|\omega_{max} - \omega_{t-1}| & if\ \vec{\alpha} > 0 \\ \omega_{t-1} + \alpha|\omega_{min} - \omega_{t-1}| & if\ \vec{\alpha} < 0 \end{cases} \tag{26}$$

which means that we are growing only a percentage of what is left within the window limits, assuring the window will never grow beyond them.

3.4. Time Shift

The time shift is a relevant parameter in sliding window approaches, as it defines two relevant properties: segment overlap and the time between each classification. Selecting an appropriate value might present itself as an easier task than with the size parameter. However, as previously stated, we hypothesize that adjusting the time shift can optimize the segmentation process, speeding up the classifier and reducing the redundancy and adjusting segment overlap accordingly. Let us consider the time shift Δ limits as defined in equation (27), which is a function of the acquisition frequency f.

$$\underbrace{\frac{1}{f}}_{\Delta_{min}} < \Delta < \underbrace{f}_{\Delta_{max}} \tag{27}$$

We will explore three different approaches, which are tested separately and are again based on the values of the entropy:

1. **Adapt1-**Δ: When entropy is high, we want to apply short time shifts. This approach aims at an exhaustive exploration of the data, by augmenting the number of analysed samples per second. Although we recognize that increasing the number of samples in degenerate data samples will naturally increase the number of misclassified samples, we expect true positive results to be in greater number, resulting in a better overall accuracy ratio. The

proposed formulation for this first approach, is as follows:

$$\triangle_{t+1} = \frac{\omega_t - (h_t * \omega_t)}{f} \tag{28}$$

where f stands for the sampling frequency, h_t the entropy at instant t and w_t the current window size measured in samples.

2. **Adapt2-**Δ: During action class state transitions, entropy values tend to be higher. In this case, we hypothesize that forwarding the window to a time period where the new action is already well defined can reduce the number of false positive results. Hence, we want to extend the time shift to its maximum value, thus yielding a minimum successive window overlap. Therefore, we propose the following formulation, which reflects our idea:

$$\Delta_{t+1} = \frac{\omega_t - ((1 - h_t) * \omega_t)}{f}. \tag{29}$$

3. **Adapt3-**Δ: We also consider interesting to study another approach when in the presence of action transitions, but addressing entropy when it becomes a volatile signal, i.e. when it experiences big differences in consecutive computed values, which is reflected in its first derivative, as is illustrated in Figure 6. Hence, to overcome this volatility effect, we consider the formulation in equation (29), integrating the 1^{st} order backward difference for the entropy signal, which results in:

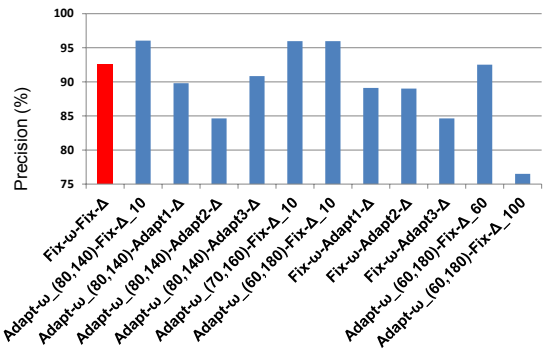
$$\Delta_{t+1} = \begin{cases} \frac{\omega_t - ((1-\nabla H) * \omega_t)}{f} & , \nabla H \geq \rho \\ \frac{\omega_t - ((1-h_t) * \omega_t)}{f} & , \nabla H < \rho \end{cases} \tag{30}$$

where $\nabla H = h_t - h_{t-1}$ corresponds to the 1^{st} order backward difference, and ρ a pre-defined numerical threshold.

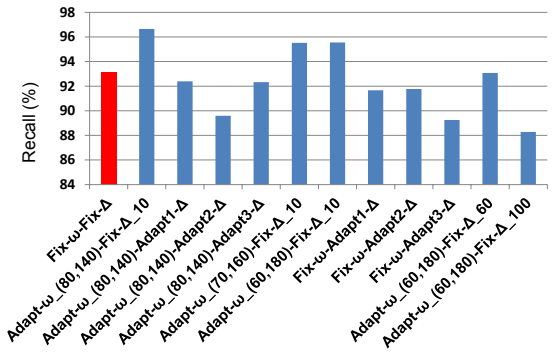
3.5. Experimental Results

To evaluate the effects of the two mentioned parameters (Window size and time shift), with respect to the proposed approaches, the classification results of different combinations are presented using two different measurements. Precision measures the number of correctly classified samples, i.e. the model accuracy, and is given by:

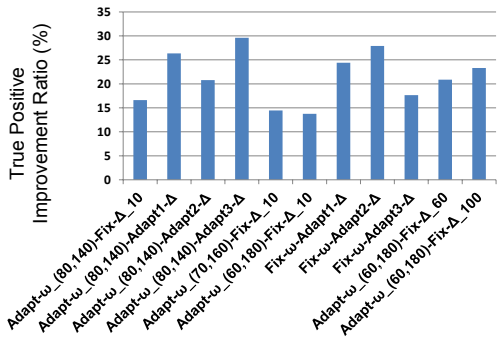
$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{31}$$



(a) Precision



(b) Recall.



(c) Accuracy Improvement (%).

	W	Ru	S	Ri	F	No
Walking	1	0	0	0	0	0
Running	0.16	0.84	0	0	0	0
Sitting	0	0	0.97	0	0.03	0
Rising	0	0	0	0.98	0	0.02
Falling	0	0	0.03	0	0.97	0
No Move	0	0	0	0	0	1

(d) Confusion matrix.

Figure 5: **(a)** Precision and **(b)** Recall measures for the different scenarios, **(c)** classification confidence improvement ratio (%) with respect to fixed approach and **(d)** per-frame classification accuracy for the approach showing the highest precision, Adapt-ω_(80,140)-Fix-Δ_10.

Precision is mostly used together with Recall, which represents the number of relevant classifications within all the results yielding a given class, such that:

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{32}$$

We have essayed different values for the adaptive parameter approach, where combinations are enumerated as defined in Section 1.4. As the bar charts in Figure 5a and Figure 5b show, adapting the window size while maintaining the time shift improves result precision when compared to the fully fixed parameter approach. Within the same strategy, modifying the thresholds for the window size does not affect precision significantly, converging to about 95% in all tested value ranges.

In the scenarios where the time shift is adaptable, the average precision is lower when compared to the previous case. The observed precision results have different justifications. For Adapt1-Δ, action transitions are characterized by small windows and time shifts, which favours a high number of misclassified samples. Small amounts of data somehow represent partially visible segments which are likely to generate confusion and contribute negatively to the classifier precision. In the Adapt2-Δ, we try to avoid transition segments by forwarding the window. Despite reducing the amount of false positives during these transition periods (because of the fast forwarding to greater confidence regions), they do exist and while the window parameters do not stabilize, the classifier may take a little longer to re-converge to the correct action. During this re-convergence procedure, misclassified samples accumulate negatively in the precision indicator. The third approach Adapt3-Δ attempts to mitigate the volatility in the entropy time series, but interestingly it contains a little of each effect of the previous two approaches. However, these effects are not so strongly visible as they are in each of their original approaches.

The window size exhibits some advantages when analysing the precision indicators, while adjusting different time shifts showed to have a more positive impact in terms of decision confidence. Figure 5c presents the improvement in classification decision confidence. The vertical axis values represent the ratio between the average confidence in each of the adaptive approaches, when compared with the fixed strategy. It is visible that all approaches are successful in improving model confidence, but the ones using an adaptive time shift improved further than the remaining. This shows that the adaptive strategy allows the classifier to estimate more confidently. The main reason for the adaptive time shift to be better than a fixed approach, is justified because when the classifier uncertainty is increasing, the time shift increases allowing the classifier to skip those areas where the outlier samples are dominant.

To complement the presented precision results in Figure 5a, Table 5d shows the confusion table with the per-frame classification amongst all available classes. Adapting the sliding window parameters has shown highly precise

results, with an overall ratio of 95%, which is an improvement with respect to our previous fixed approach [2], which is depicted in a red bar in Figures 5a and 5b. We conclude the section with Figures 7 and 8 (at the end of this manuscript), where we can see an action sequence, the ground truth annotation and the corresponding delay and classified classes.

3.5.1. Anticipating the Recognition of Actions

One other relevant factor is the convergence speed, i.e. how long it takes for the classifier to detect the correct event after it actually started. In these experiments we count the number of missed frames in the classification process until the correct decision is achieved, i.e. the number of misclassified frames between the ground truth annotation and the actual model classification. This effect is specially felt on action transitions, where the model needs to re adjust the classified state from one action class to a different one. Figure 6 illustrates the differences between using fixed and the adaptive time shift approach. We can see that without adaptive time shift in Figure 6a the correct decision is achieved around frame no. 210, whereas when using the proposed approach, that decision is anticipated to frame no. 170, where the ground truth is marked about frame no. 155. With this example we aim to demonstrate that we can anticipate the convergence to the correct class with respect to ground truth annotation. The Bayesian nature of the classifier will show some resistance to this change, due to the effect of the prior probability, which naturally delays the state transition. Figure 6c shows that most of the approaches improve the convergence speed particularly the approaches belonging to adaptive window size with fixed time shift. We can see that some variations reduce the delay in almost 70% with respect to fixed width approach, whereas our best approach (Adapt-ω_(80,140)-Fix-Δ_10) in terms of precision and recall, also reveals itself to be the best in terms of speed improvement. In terms of segmentation accuracy, it means that segments will be labelled much more accurately, due to the fact that model classification decisions tend to be closer to their ground truth markers.

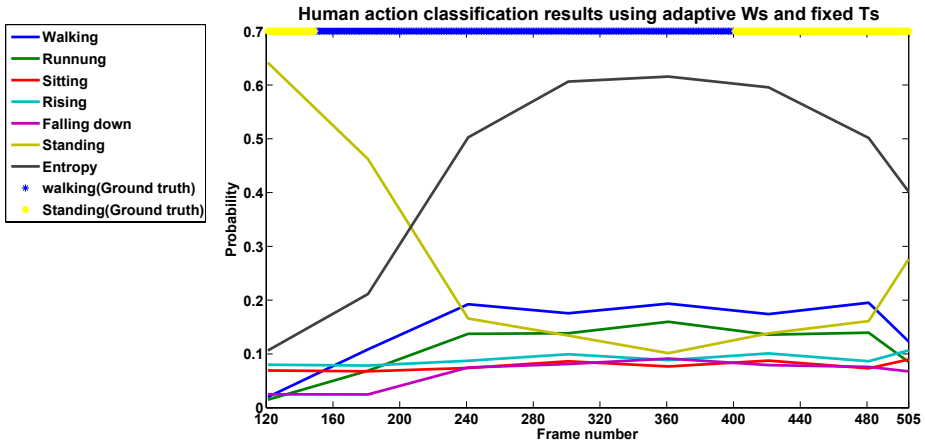
3.5.2. Result Discussion

The results present in the previous sections allow us to observe that the different approaches have a different impact in the model precision, confidence and speed with which the model decides with respect to a given action class. The Table 4 presents a summary of the effects that both window parameters have in the different analysed indicators.

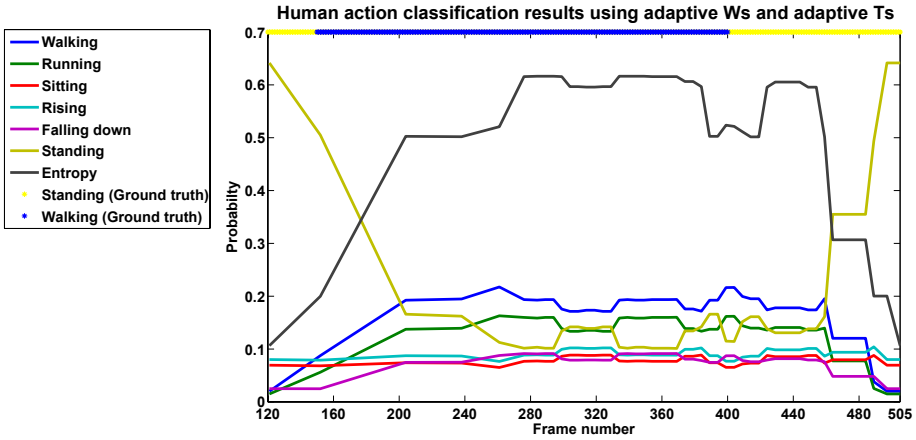
Table 4: Comparison of how the window parameters are affected amongst the different proposed approaches.

Measure	Description
Precision	adaptive ω_s with fixed shorter Δ_t , ω_s's thresholds have less effect
Confidence	adaptive Δ_t, especially 2nd and 3rd Δ_t approaches
Anticipation speed	adaptive ω_s with fixed shorter Δ_t, ω_s's thresholds have less effect

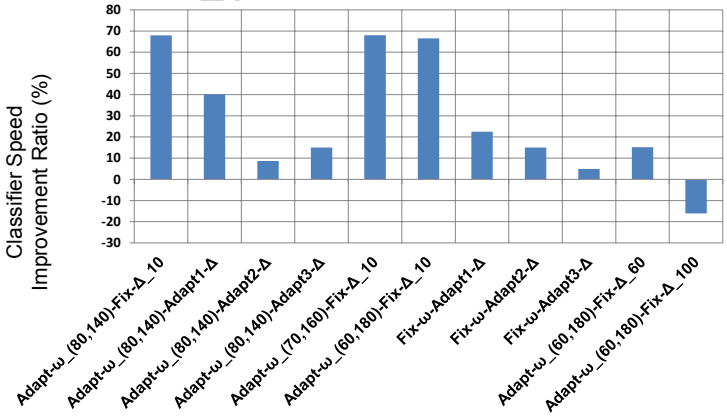
The adaptive behaviour of the sliding window shows that it can improve all mentioned important classification



(a) Fixed Time Shift



(b) Adaptive Time Shift



(c) Convergence speed ratio improvement.

Figure 6: A sample results of adaptive sliding window approach using fixed and adaptive time shift approach (The coloured top bar of the frames show the ground truth, the black line shows the entropy signal). Convergence speed Improvement (Percentage) with adaptive approaches when compared to fixed approach delays.

indicator outputs. However, the selection of the most appropriate strategy is highly dependent on the main goal we want our classification framework to achieve. Adapting the window size is more beneficial to the precision and recall indicators. On the other hand, adapting the time shift has higher impact on confidence and in the anticipation of the decision of the classifier, bringing the correct decision closer to the ground truth instant, consequently providing a more accurate temporal segmentation. Although it is clear that different parameters impact indicators differently, our experimental results also show that it is possible to find a good compromise between all indicators, as we demonstrate with the approach Adapt-ω(80,140)-Adapt1-Δ.

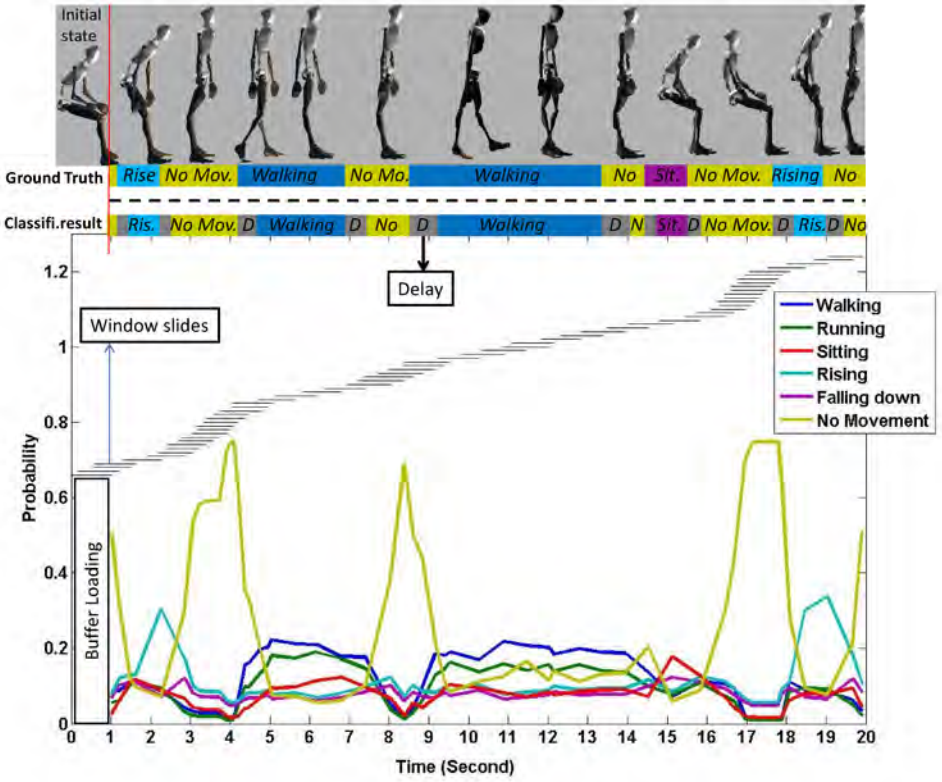
It is also relevant to mention that having shorter time shifts tends to increase the computational cost of the classification process because of the higher rate of classifications per second. Thus, there is a trade off between the amount of time shift and the computational cost, which is where the adaptive time shift approach can also play a relevant role.

When comparing our work with other approaches, we can point the following main advantages: 1) our approach does not rely on the type of signal, devices or processing algorithms [16]; 2) it is applied beyond data mining processes [17]; 3) the adaptable parameter approach is applied to classification processes beyond the selection of good learning segments, and the adaptive process demonstrates to improve classifier performance [16, 17]; 4) Because of its complete abstraction, it can be easily integrated with any classification process which uses a sliding window approach.

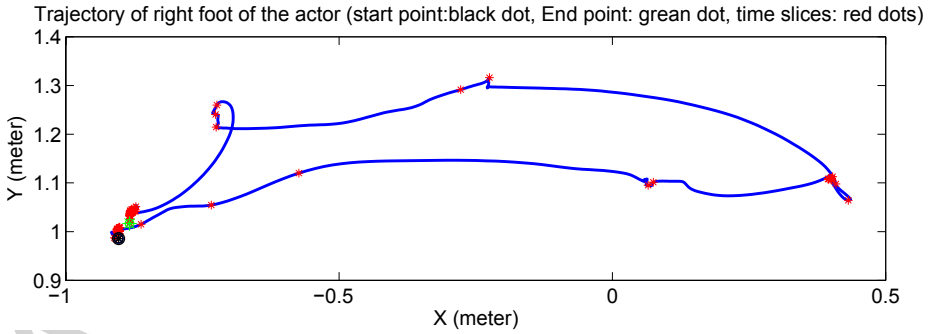
4. Conclusions and Future Work

In this paper we propose a solution to action classification, an adaptive approach to continuously adjust the two key parameters in sliding windows: size and time shift. We have demonstrated that changes in these parameters have a high impact in the model learning. We have posed this as an entropy minimization problem, formulating a feedback model, which based on entropy and previous sliding window parameters, allowed the window to continuously adapt itself to the classification process. We have tested numerous scenarios, which used different values for the limits of each parameter, and successfully demonstrated our approach to improve results, verified through adequate classification metrics: precision, recall, confidence and convergence time (measured in frames). Moreover, our formulation is generalizable, i.e. it can be applicable to abstract classification frameworks, as long as they are based on the sliding window paradigm and values for entropy and window parameters are available.

Our future work encompasses the extension of our research to an accurate selection of window parameter limits. We expect to obtain generalizable limit selection, which can be applied in general classification problem in which a set of variables is known. We will also direct our attention into the development of an abstract classification framework, based on the proposed adaptive paradigm, for MatLAB platform.

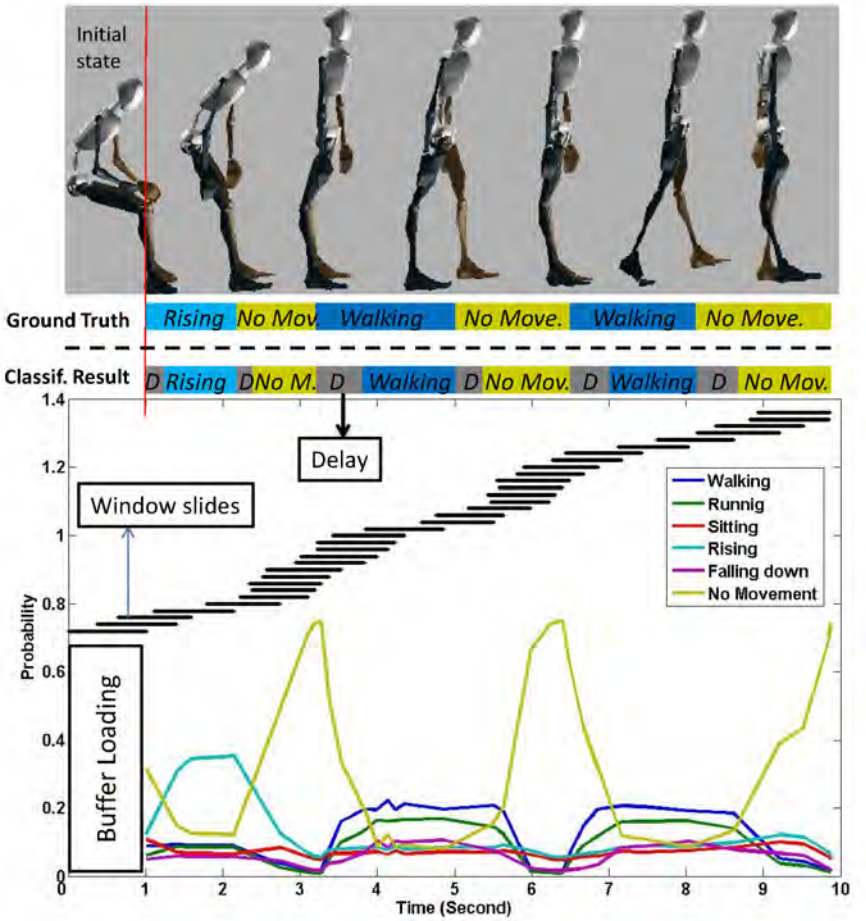


(a) Example1

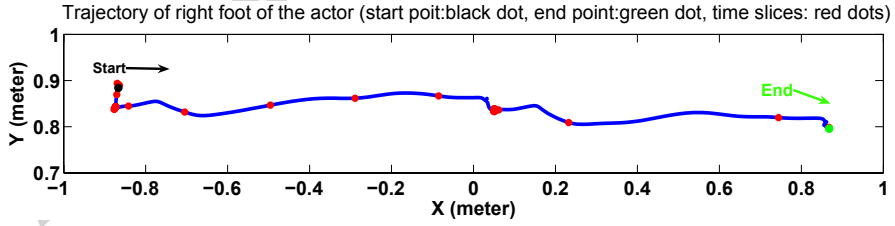


(b) Trajectory for Example 1

Figure 7: Sample sequence: a person starts in a rest position, rises and walks around, and returns to the initial state. Gray areas in the classification bar symbolize the delay between ground truth and classified action states. The output distribution with each action probability at a given instant is presented in the graph. The bottom graph represents the trajectory of the performed activity sequence. The sequence is classified using the proposed adaptive sliding window approach and samples at a frequency of 120Hz. The body models presented on top of figure (a) belong to the XSENS Software Development Kit.



(a) Example2



(b) Trajectory for Example 2

Figure 8: Sample sequence: a person starts in a rest standing position and makes a stretch run until (s)he stops. The image structure and conditions are the same as the previous Figure 7. The body models presented on top of figure (a) belong to the XSENS Software Development Kit.

Acknowledgements

This work has been supported by Institute of Systems and Robotics from University of Coimbra, Portugal, and Khalifa University, Abu Dhabi, UAE. Luís Santos and Kamrad Khoshhal are supported by FCT - Portuguese Foundation for Science and Technology, Grants # 65935/2009 and # 70640/2010 respectively.

References

[1] K. K. Rousposhti, J. Dias, Improved semantic-based human interaction understanding using context-based knowledge, in: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013), 2013, pp. 2899–2904. doi:10.1109/SMC.2013.494.

[2] K. Khoshhal, J. Dias, Probabilistic human interaction understanding - exploring relationship between human body motion and the environmental context, Pattern Recognition Letters Special Issue on Scene Understandings and Behaviours Analysis (2013) 820–830. doi:http://dx.doi.org/10.1016/j.patrec.2012.09.021.

[3] L. Santos, J. Prado, J. Dias, Human robot interaction studies on Laban Human Movement Analysis and dynamic background segmentation, in: IROS 2009, The 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems, 2009.

[4] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding 115 (2) (2011) 224–241.

[5] L. Zelnik-Manor, M. Irani, Event-based analysis of video, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 2, 2001, pp. II–123 – II–130 vol.2.

[6] Z. Feng, T.-J. Cham, Video-based human action classification with ambiguous correspondences, in: Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on, 2005, p. 82.

[7] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 1, 2005, pp. 166 – 173 Vol. 1.

[8] Y. Ke, R. Sukthankar, M. Hebert, Event detection in crowded videos, in: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, 2007, pp. 1 –8.

[9] L. Zelnik-Manor, M. Irani, Event-based analysis of video, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, 2001.

[10] H. Zhong, J. Shi, M. Visontai, Detecting unusual activity in video, in: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, Vol. 2, 2004, pp. II–819 – II–826 Vol.2.

[11] Z. Feng, T.-J. Cham, Video-based human action classification with ambiguous correspondences, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005.

[12] T. Darrell, A. Pentland, Space-time gestures, in: Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on, 1993, pp. 335 –340.

[13] P. Morguet, M. Lang, Spotting dynamic hand gestures in video image sequences using Hidden Markov Models, in: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, 1998, pp. 193 –197 vol.3.

[14] A. Bobick, Y. Ivanov, Action recognition using probabilistic parsing, in: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, 1998, pp. 196 –202.

[15] A. Wilson, A. Bobick, Parametric Hidden Markov Models for gesture recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 21 (9) (1999) 884 –900.

[16] Laguna, J. Ortiz, Olaya, A. García, D. Borrajo, A dynamic sliding window approach for activity recognition, in: Proceedings of the 19th international conference on User modeling, adaption, and personalization, UMAP'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 219–230.
URL http://dl.acm.org/citation.cfm?id=2021855.2021875

[17] Yang, Yi, Mao, Guojun, A self-adaptive sliding window technique for mining data streams, in: Z. Du (Ed.), Intelligence Computation and Evolutionary Computation, Vol. 180 of Advances in Intelligent Systems and Computing, Springer Berlin Heidelberg, 2013, pp. 689–697. doi:10.1007/978-3-642-31656-2_93.
URL http://dx.doi.org/10.1007/978-3-642-31656-2_93

[18] S. Vitaladevuni, V. Kellokumpu, L. Davis, Action recognition using ballistic dynamics, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1 –8.

[19] C. Lu, H. Liu, N. Ferrier, Multidimensional motion segmentation and identification, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, Vol. 2, 2000, pp. 629 –636 vol.2.

[20] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, N. S. Pollard, Segmenting motion capture data into distinct behaviors, in: Proceedings of Graphics Interface 2004, GI '04, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004, pp. 185–194.

[21] T. Abbas, B. MacDonald, Robust trajectory segmentation for programming by demonstration, in: Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on, 2009, pp. 1204 –1209.

[22] P. Banerjee, R. Nevatia, Dynamics based trajectory segmentation for UAV videos, in: Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on, 2010, pp. 345 –352.

[23] S. Rao, R. Tron, R. Vidal, Y. Ma, Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (10) (2010) 1832 –1845.

[24] Q. Shi, L. Wang, L. Cheng, A. Smola, Discriminative human action segmentation and recognition using semi-Markov model, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, 2008, pp. 1 –8. doi:10.1109/CVPR.2008.4587557.

[25] A. S. Ogale, A. Karapurkar, G. Guerra-filho, Y. Aloimonos, View-invariant identification of pose sequences for action recognition, in: In VACE, 2004.

[26] D. Weinland, R. Ronfard, E. Boyer, Automatic discovery of action taxonomies from multiple views, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, 2006, pp. 1639 – 1645.

[27] F. I. Bashir, et al., View-invariant motion trajectory-based activity classification and recognition (2006).

[28] F. Lv, R. Nevatia, Recognition and segmentation of 3-d human action using HMM and multi-class Adaboost, in: Proceedings of the 9th European conference on Computer Vision - Volume Part IV, ECCV'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 359–372. doi:10.1007/11744085_28.

[29] P. Peursum, S. Venkatesh, G. West, Tracking-as-recognition for articulated full-body human motion analysis, in: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007, pp. 1 –8.

[30] S. B. Kang, K. Ikeuchi, Towards automatic robot instruction from perception-temporal segmentation of tasks from human hand motion, Robotics and Automation, IEEE Transactions on 11 (5) (1995) 670 –681.

[31] V. Pavlovic, J. Rehg, Impact of dynamic model learning on classification of human motion, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, Vol. 1, 2000, pp. 788 –795 vol.1.

[32] D. O. Trevor, T. Hastie, M. Black, Functional analysis of human motion data (2000).

[33] K. Kahol, Gesture segmentation in complex motion sequences, Master's thesis, Arizona State University (2003).

[34] K. Oka, Y. Sato, H. Koike, Real-time fingertip tracking and gesture recognition, Computer Graphics and Applications, IEEE 22 (6) (2002) 64 – 71.

[35] L. Shao, X. Zhen, D. Tao, X. Li, Spatio-temporal Laplacian Pyramid Coding for action recognition, Cybernetics, IEEE Transactions on PP (99) (2014) 1–1. doi:10.1109/TCYB.2013.2273174.

[36] L. Shao, S. Jones, X. Li, Efficient search and localization of human actions in video databases, Circuits and Systems for Video Technology, IEEE Transactions on 24 (3) (2014) 504–512. doi:10.1109/TCSVT.2013.2276700.

[37] L. Liu, L. Shao, P. Rockett, Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition, Pattern Recognition 46 (7) (2013) 1810 – 1818. doi:http://dx.doi.org/10.1016/j.patcog.2012.10.004.
URL http://www.sciencedirect.com/science/article/pii/S0031320312004372

[38] K. Khoshhal, H. Aliakbarpour, J. Quintas, P. Drews, J. Dias, Probabilistic LMA-based classification of human behaviour understanding using power spectrum technique, in: 13th International Conference on Information Fusion, 2010.

[39] K. Khoshhal, L. Santos, H. Aliakbarpour, J. Dias, Parameterizing interpersonal behaviour with Laban Movement Analysis - a Bayesian approach, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012. doi:10.1109/CVPRW.2012.6239349.

[40] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (7) (1936) 179–188.

[41] C. R. Rao, The utilization of multiple measurements in problems of biological classification, Journal of the Royal Statistical Society - Series B 10 (2) (1948) 159–203.