Systematic study on dimensionality reduction in the gesture phase segmentation problem

Victor G. O. M. Nicola
University of São Paulo
São Paulo, Brazil
victor.nicola@usp.br

Renata C. B. Madeo
Regional Federal Court of the 3rd Region
São Paulo, Brazil
renata.madeo@gmail.com

Sarajane M. Peres
University of São Paulo
São Paulo, Brazil
sarajane@usp.br

***Abstract*—In this paper, we present the results obtained in a systematic study related to dimensionality reduction on gesture windowed data. The aim of the study was to analyze the effects that such reduction causes on the performance of classification models used to implement the gesture phase segmentation task. Piecewise Aggregate Approximation was used to implement the dimensionality reduction and k-Nearest Neighbors was used to implement the classification models. The results showed that the dimensionality reduction can improve the classification models' performance both by decreasing the complexity of their decision space and by improving the quality of the data.**

***Index Terms*—Gesture Phase Segmentation, Gesture Analysis, Dimensionality Reduction, Piecewise Aggregate Approximation, k-Nearest Neighbors**

I. Introduction

Automating the gesture interpretation allows developing applications with interaction ability close to natural human interaction [1], [2]. There are initiatives that deal with the gesture recognition in finite vocabulary scenarios of less complexity (human-robot interaction or implementation of interfaces based on touchscreen resources) or greater complexity (oral and sign language processing) [3]. Another important scenario is the understanding of natural gesticulation, in which there is no previously established finite vocabulary of gestures, but there is a gestural behavior capable of carrying information. This type of gesture is studied by the gesture theory.

Gesture theory seeks to understand human gestures and how they can transmit information. This is not restricted to manual gestures, although they are the most important channel for transmitting non-verbal information. According to D. McNeill [4]: "gesture are not just the arms waving in the air, but symbols that exhibit meaning in their own right". Gestures usually accompany the speech given in oral language and support both the construction of the speech and its understanding. Reviews on automatic analysis of manual gestures and their applicability are found in [5]–[7].

Gesture theory provides frameworks and some level of formalization that allows you to automate the interpretation of gestures. One of these frameworks refers to the gesture phases [8], [9], the focus of interest in this paper. Gesture phases are a hierarchy of movements that composes or describes human gesticulation. This hierarchy divides gestures into segments, or phases, called preparation, stroke, hold and retraction. To develop applications capable of using the information in the gesture phases, it is first necessary to automate the gesture phase segmentation task. This automation assumes that there is a speech being uttered by a person and recorded on video. This video is input to a computer system capable of segmenting it into sections according to each phase of the gesture.

From the automation standpoint, it is necessary to represent the gestures so they are interpretable by computers. A way is to capture the position of the hands during gesticulation, organize them sequentially into a type of multi-dimensional time series and submit it to a classification model capable of identifying which phase each value in that series belongs to. The key issue in this process is that temporal information needs to be considered when it comes to gesture analysis. One way to work with this information is through the data windowing [10]. However, data windowing can create high dimensional representations. In addition, information extracted from a video will contain noise. These two characteristics make it difficult to create good classification models. Dimensionality reduction and signal smoothing can help in this scenario.

This paper presents a systematic study on the effect of applying a dimensionality reduction method (Piecewise Aggregate Approximation) on time series derived from the application of the sliding window method on data related to natural gestures. To evaluate such effects, the gesture phase segmentation task was solved using the k-Nearest Neighbors classification algorithm. PAA and k-NN were chosen due to their low modeling complexities, which allows to minimize exogenous influences that complex methods could bring.

This paper is organized as follows: Section II presents basic concepts related to this study; Section III defines the gesture phases classification problem; Section IV describes the protocol followed to systematize the experiments and support the analysis; Section V discusses the results and analyzes the trends observed; and Section VI summarizes the main findings.

II. Theoretical background

A. Gesture Theory

In gesture theory, the gesture phase segmentation task analyzes the gesture movement hierarchy, allowing to relate it to discourse units [8]. A framework proposed by A. Kendon [8]

The authors thank Tutorial Education Program of Brazilian Education Ministry (PET/Sesu/MEC - Brazil).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

defines: *rest position*, a relaxed position without movement; Gesture Unit (*G-unit*), the period of time from when hands leave rest position to the moment they return; and Gesture Phrase (*G-phrase*), a component of a G-unit, composed by one or more gesture phases. The gesture phases are: *preparation*, when hands move to the position where the movement will be performed; *pre-stroke hold*, a pause between preparation and stroke; *stroke*, the movement that conveys meaning to the gesture; *post-stroke hold*, a pause between stroke and retraction; *retraction*: hands move back to rest position.

To perform automated gesture phase segmentation, it is first necessary to obtain videos of people gesticulating and label segments with its corresponding phase. This process is usually referred as coding. As coding is a subjective task, [9] focus on proposing a grammar for gesture phase segmentation that increases inter-coder reliability, helping coders to analyze gesticulation. The main difference between previous framework and the latter [9] is that the expressive phase can either contain a stroke that may be surrounded by *dependent holds* (such as the framework above), or contain an *independent hold* that is an expressive phase with no movement that replaces a stroke.

There are two major issues regarding automated gesture analysis segmentation [11]: the similarity between holds and rest positions, when there is almost no movement; and the difficulty to precise a boundary between transitional phases, such as preparation or retraction, and rest positions. Due to this subjectivity, coders may differ on the labeling assigned to video segments close to the phase transitions (see [11]). Due to the difficulties raised, the automation of gesture phase segmentation is commonly performed with the phases: rest position, preparation, stroke, hold and retraction (see Figure 1).

B. Classification tasks and k-NN algorithm

In the context of this study, the segmentation problem is solved by modeling the well-known data analysis task called "classification" [12]. Classification tasks can be modeled as binary or multiclass classification. The former considers that only two classes are involved in the problem; the latter considers several classes. To act on this type of task, supervised learning algorithms are usually applied. We choose the k-Nearest Neighbors (k-NN) algorithm to carry out this study. It implements a supervised lazy learning method based on instances [12]: its training is the storage of a labeled dataset; the test (or the classification of a new element) is based on calculating the similarity between the stored data points and the unlabeled data point for which a label is to be determined; the k-most similar data points are recovered; the class assigned to the unlabeled data point is determined by the most frequent label among those assigned to the data points recovered.

C. Sliding Windows

The gesture phase segmentation problem involves the analysis of information distributed over time. There are two alternatives to solve this type of problem: a strategy of analysis capable of implementing temporal reasoning [13] or a strategy that embed temporal information into the vector representation of the data [14]. Following the strategy used in our previous works, we chose the second option for this study. This option involves modeling time information through sliding windows.

Sliding windows are commonly used to deal with time series problems as it brings a temporal aspect to each data [10]. It consists in representing each data point using features of the data point itself and features of the previous and/or later data points (depending on the task to be achieved). For instance, given a time series T = (t_1, ... t_i, ... t_n), using a sliding window with size *w*, t_i could be represented using features from ($t_{i\text{-}w}$, $t_{i\text{-}w+1}$, ... t_i), if only previous information must be used.

However, as discussed in [15], it may be difficult to find an ideal window size, since there is a trade-off between window size (the larger the window size, the more data it covers) and recognition performance. In addition, depending on the chosen window size, the dimensionality of the vector space in which the data analysis algorithm will act can become large enough to impact the efficiency of the task resolution.

D. Dimensionality Reduction

The notion of dimensionality is associated with the number of features used to describe an observation or a data point. The more features are used, the greater the dimension of the decision space associated to the classification problem, the more operations are required to implement the algorithm that will solve the problem and the stronger is the impact on processing time. In order to minimize this problem, we choose to apply the reduction technique called Piecewise Aggregate Approximation (PAA) [16]. PAA is a simple method in which the original series of size n is reduced to a series of size N, following $x_i = (N/n) \sum_{j=(n/N)(i-1)+1}^{(n/N)i} x_j$, in which each element x_i of the reduced series is given by averaging a subset of points x_j in the original series. The size of the new series can be defined as a percentage of the size of the original series.

III. Problem Definition: Gesture Phases Segmentation

The representation of gestures in video form is described as a sequence $S = \{f_1, f_2, ..., f_N\}$ of RGB image frames. Thus, to classify the gesture phases, each frame must receive the label according to the phase in which the hands are in, i.e., the classifier must be able to receive f_i from S as input, and classify it according to one of the classes in $C = \{0, 1\}$ for binary tests or $C = \{0, 1, 2, 3, 4\}$ for multiclass classifications.

Binary classification tests consist of determining whether the hands in a given frame are at rest or in the middle of the gesture unit. In the multiclass tests, five classes were used: 0 for rest, 1 for preparation, 2 for stroke, 3 for hold, 4 for retraction. Both post-stroke hold and pre-stroke hold were considered to be hold, due to difficulties in labeling.

Since each frame is classified according to the desired classes, the problem of gesture phase segmentation is reduced to scanning the sequence of frames now labeled, identifying sequences of frames that have the same label. Then, each sequence of frames is considered a segment referring to the gesture phase associated with the label of its frames.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

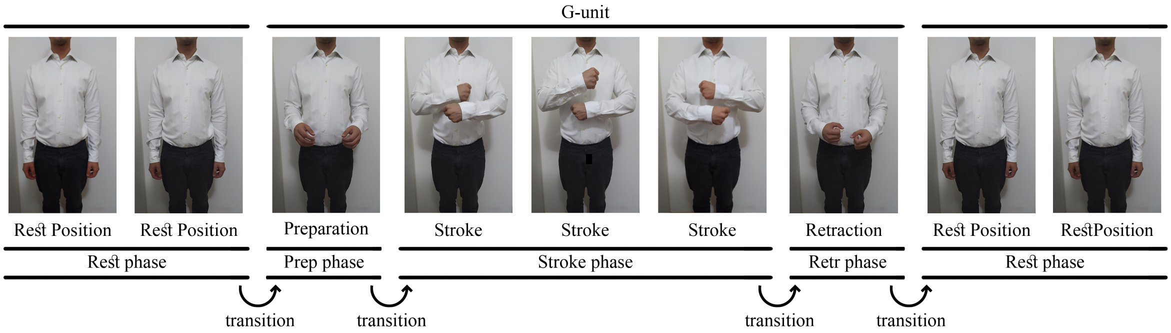

Fig. 1: Gesture phases examples: hold phases may appear instead of stroke phases

IV. Experiment protocol

A. Dataset

Gesture Phase Segmentation Dataset[1] is a dataset extracted from seven videos of people telling stories while gesticulating [3]. There are three people (A, B, C) telling three different stories (1, 2, 3). We have selected videos A1 (i.e., person A telling story 1), A3, B1, B3, C1 and C3[2].

According to [3] and [17], Microsoft Kinect™ Sensor was used to obtain, for each frame, the coordinates (x, y, z) of hands and wrists, which were normalized using the position of head and chest, also obtained by the sensor. The normalized data were used for calculating speed (scalar quantity) and velocity (vector quantity) and acceleration of hands and wrists. Labeling was carried out manually by people analyzing RGB images of each frame and using the grammar proposed in [9].

Table I shows the distribution of gesture phases among frames of the selected videos, following the labeling provided with the data. According to [11], the labeling of gestures phase is a subjective task, mainly with regards to the transition frames between phases. Thus, these authors carried out a re-labeling to this dataset, with the participation of two new coders, to perform an agreement analysis. For this dataset, they found low divergence between the labelings, getting Krippendorff's Alpha coefficients around 0.8. This threshold indicates that there is substantial or perfect agreement between coders [18], meaning that any of the labelings can be used to train classification models.

TABLE I: Dataset description

Phases	**A1**	**A3**	**B1**	**B3**	**C1**	**C3**
Rest	698	662	74	194	286	362
Preparation	163	279	411	469	236	338
Stroke	656	535	287	390	262	389
Hold	39	150	217	201	193	144
Retraction	191	208	84	170	134	215
Total frames	**1747**	**1834**	**1073**	**1424**	**1111**	**1448**

B. Types of error

Segmenting gestural units implies using binary classifiers and segmentation of gesture phases implies using multiclass classifiers. In this work, the performance of the binary classifiers were evaluated using the classic measure of F-score [19]. Multiclass classifiers were evaluated using the measure of accuracy (rate of frames classified correctly).

However, from the point of view of an expert in the Linguistic field, the errors that occur in the transitions between phases can be considered less important [3]. Therefore, the evaluation of segmentation results should include an analysis of special errors. To accomplish this need, three types of special errors were calculated based on [3]:

- *Irrelevant Transition Error* (ITE): it is an error that occurs at the edge of the transition between two phases. In this case, the classifier shifts the segmentation by classifying frames from the current phase as being frames from the next phase or vice versa. Such errors, as long as they involve few frames, can be considered irrelevant.
- *Serious Transition Error* (STE): it is an error that also occurs at the edge, however the phase indicated by the classifier is different from the phases involved in the transition. This is an important error, as it inserts a gesture phase in a place where it does not exist.
- *Internal Error* (IE): it is an error that occurs outside the edge of the transition. This error is important for the same reason that STE is.

Figure 2 illustrates the special errors: each occurrence of a geometric symbol means the phase label assigned to a frame in the video; the transition edge has eight frames, four frames before and four after the transition between phases.

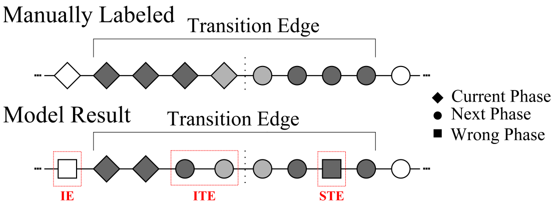

Fig. 2: Graphical representation for the special errors

[1] https://archive.ics.uci.edu/ml/datasets/gesture+phase+segmentation

[2] Videos referring to person 2 have few frames related to the resting phase, which hindered the tests related to the G-units. For this reason, the videos referring to that person were not considered in the experiments reported here.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

C. Experiment setup

The dataset used allows the combination of different features to describe the movements that make up the gestures. We have divided the possible features into two subsets, which characterize two experimentation environments:

- *Environment #A*: features that describe movement trajectories in the real-world bi-dimensional space using information from the x, y-coordinates of the hands.
- *Environment #B*: features that characterize the movement performed in the space-time dimension using the velocity information imposed on the movement.

Each of the environments was used for the study on the G-unit segmentation (binary classifier - *Experiment #1*) and for the study on the gesture phase segmentation (multiclass classifier - *Experiment #2*). For each environment we built different combinations of features and the following steps:

- *Feature selection step*: the x, y-coordinates (*Environment #A*) and the velocities in $\{(x, y)\}$ (*Environment #B*) were used considering only the right hand, only the left hand and both hands.
- *Windowing step*: the frames were grouped in windows of pre-established sizes (from 1 to 10 frames with 1 for step size, and 10 to 100 frames with 10 for step size)[3]. Each window became a data point labeled with the same label as the central frame of the window.
- *Rearrangement step*: the windowed data was reorganized so that the values of each feature were sequentially placed inside the window. This rearrangement was necessary to perform the dimensionality reduction step in each axis of the original vector space.
- *Dimensionality reduction step*: PAA was applied to the series of values formed for each feature within the window, considering a pre-established reduction rate (70% to 10% with 10 for step size).
- *Segmentation step*: the classifiers were trained and evaluated. The k-NN was performed for k ranging from 2 to 10 with 1 for step size and Euclidean distance.

Figure 3 illustrates the steps performed in each environment. In this figure, we represent three frames from the original video, two data points constructed through the windowing and rearrangement procedures, the dimensionality reduction for these data points and the gesture phases segmentation produced using the k-NN classifier (the first frame in the "rest phase" - above the dotted line - and the second and third frames in the "preparation phase" - below the dotted line).

Considering the above, the total set of experiments comprised 109,440 instances of classification models. Figure 4 summarizes this information. All models were created under the holdout approach, using $2/3$ of a video for training (the initial frames of the video) and $1/3$ of the video frames for testing (the final frames of the video). This procedure ensured the presence of all class labels involved in the classification problem, both in the training fold and in the test fold.

All codings were implemented using MATLAB®. The k-NN logic was implemented with the support of Statistics and Machine Learning Toolbox. The PAA logic was implemented with the support of the codes published by [20], [21].

V. Results analysis

This section contains the synthesis and analysis of the results obtained with the experiments. It is divided into three parts: k-NN general performance, analysis of results obtained in the *Experiment #1*, analysis of results obtained in the *Experiment #2*. In general, the best classification rates were obtained using the data representation based on information from both hands. Thus, most of the analysis presented herein concerns the classifiers obtained with such a representation.

A. General performance analysis of the k-NN algorithm

First, we studied the performance of k-NN to verify whether it would be robust enough to properly support tests with dimensionality reduction. Table II shows the results of k-NN in terms of average F-score for binary classifiers, and average accuracy for multiclass classifiers. Average results were calculated taking all values defined for k (cf. Section IV-C). In this table, the performance of k-NN is shown from different points of view, for each video and type of classification problem. The results are presented in terms of: higher and lower averages (μ), greater standard deviation (σ), and greater coefficient of variation (c). We argue that k-NN is appropriate to support the systematic study reported in this paper, since the low values for σ and c_v indicate small variations for the measures F-score/accuracy, considering different values assigned to k.

B. Experiment #1 - Binary classification

1) Environment #A: Figure 5 presents the average F-score considering all classifiers obtained for the same window size. using data from video A1. Each series in this graph concerns one of the three possibilities of representing a gesture using hand position information.

The analysis of these series allows us to verify the superior quality of results obtained when both hands are used to represent the gesture. The series referring to both hands reveals that the best classification performance is obtained in the window with size 10 (F-score = 0.877), with small variations from sizes 5 and 20. The series referring to the left hand positions presents greater stability than the one referring to the right, in windows with small size. However, both lead to similar performances with large windows, and lower performance than that obtained using the information from both hands.

Video A3 showed the same behavior, but with a decrease in terms of the best F-score (0.804). Videos C1 and C3 led to other behaviors: *(a - video C1)* best F-scores (≈ 0.70) obtained in windows of sizes close to 10, with similar performances for the representation based on both hands and for the representation based on the left hand; *(b - video C3)* best F-scores (≈ 0.90) obtained in windows of sizes close to 20, with similar performances for the representation based on both hands and for the representation based on the right hand.

[3] Using window size 1 means carrying out the experiment without windowing, i.e., without considering the vector representation for the time dimension.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

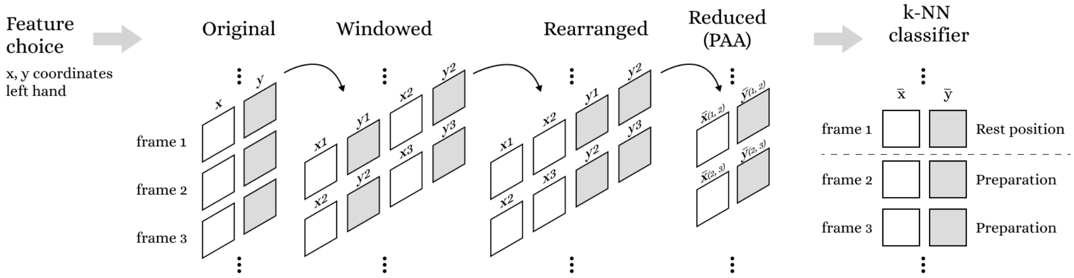

Fig. 3: Steps carried out in each environment

TABLE II: k-NN performance: k varying from 1 to 10 in original windows using information from both hands; w. refers to window size; in the first column p refers to position information and v refers to velocity information

Binary	Maximum μ				Minimum μ				Maximum σ				Maximum c_v			
Video	μ	σ	c_v	w.	μ	σ	c_v	w.	μ	σ	c_v	w.	μ	σ	c_v	w.
A1$_p$	0.877	0.004	0.004	10	0.662	0.004	0.006	90	0.845	0.045	0.053	1	0.845	0.045	0.053	1
A1$_v$	0.805	0.009	0.012	10	0.581	0.008	0.013	100	0.641	0.025	0.039	50	0.641	0.025	0.039	50
A3$_p$	0.805	0.004	0.005	10	0.753	0.005	0.007	70	0.758	0.042	0.055	1	0.758	0.042	0.055	1
A3$_v$	0.645	0.012	0.018	70	0.506	0.013	0.026	20	0.631	0.021	0.033	80	0.524	0.020	0.038	1
C1$_p$	0.733	0.043	0.059	70	0.403	0.012	0.031	100	0.602	0.094	0.155	4	0.602	0.094	0.155	4
C1$_v$	0.536	0.018	0.033	4	0.309	0.007	0.022	80	0.429	0.034	0.079	1	0.429	0.034	0.079	1
C3$_p$	0.938	0.003	0.003	20	0.680	0.016	0.023	100	0.774	0.080	0.104	3	0.774	0.080	0.104	3
C3$_v$	0.567	0.009	0.015	2	0.224	0.038	0.173	100	0.224	0.039	0.173	100	0.224	0.039	0.173	100
Multiclass	Maximum μ				Minimum μ				Maximum σ				Maximum c_v			
Video	μ	σ	c_v	w.	μ	σ	c_v	w.	μ	σ	c_v	w.	μ	σ	c_v	w.
A1$_p$	0.718	0.006	0.008	20	0.529	0.002	0.004	100	0.633	0.039	0.062	2	0.599	0.038	0.064	1
A1$_v$	0.634	0.008	0.013	7	0.415	0.006	0.015	90	0.596	0.023	0.038	3	0.458	0.021	0.046	50
A3$_p$	0.633	0.003	0.005	30	0.491	0.005	0.011	100	0.521	0.026	0.050	1	0.521	0.026	0.050	1
A3$_v$	0.542	0.016	0.030	40	0.396	0.026	0.067	1	0.418	0.030	0.072	2	0.418	0.030	0.072	2
C1$_p$	0.539	0.005	0.009	30	0.311	0.030	0.096	3	0.365	0.061	0.166	5	0.365	0.061	0.166	5
C1$_v$	0.454	0.015	0.033	10	0.233	0.047	0.199	70	0.233	0.047	0.199	70	0.233	0.047	0.199	70
C3$_p$	0.570	0.027	0.047	20	0.297	0.034	0.116	1	0.356	0.069	0.194	3	0.356	0.069	0.194	3
C3$_v$	0.367	0.018	0.049	8	0.217	0.014	0.065	100	0.234	0.029	0.125	90	0.234	0.029	0.125	90

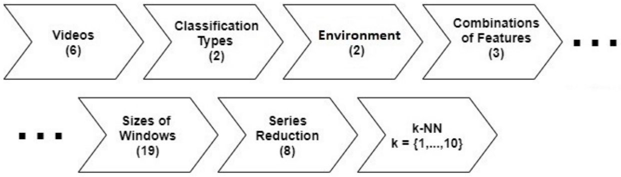

Fig. 4: Workflow for the experiments

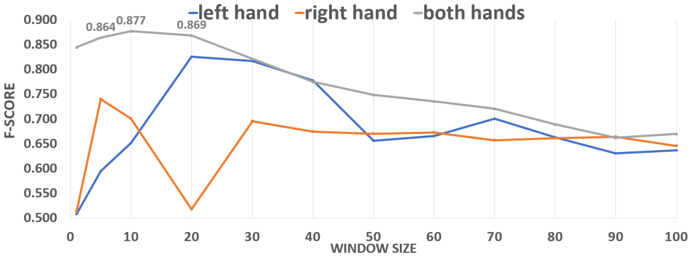

Fig. 5: Average F-score for each window size built with data from video A1 and hands' position information

From this analysis, we conclude the best performance was obtained in windows of size between 5 and 20 and representation using both hands, with performance drop for windows of sizes greater than 40. Video C1 has characteristics that make its segmentation harder for the classification model.

Figure 6 shows the results of the classification models performed on data with reduced dimensions through the application of PAA. These results refer to the video A1 considering the representation based on both hands. Each series in this graph refers to the performance of classification models trained with the original size windows and their reduced versions. This graph has four main points of attention:

- the rightmost points in each series refer to classification models built for windows without dimensionality reduction (100% of the values in the window are presented as input for the k-NN algorithm);
- the series label "Both hands - 10", for instance, indicates that the representation is based on both hands and the window has original size 10;
- scenarios with windows with original size 1 or 5 were not submitted to dimensionality reduction, so the corresponding series have one point to the right of the graph;

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

- for the other scenarios, each point in the series, from right to left, indicates that the original window has undergone a percentage reduction.

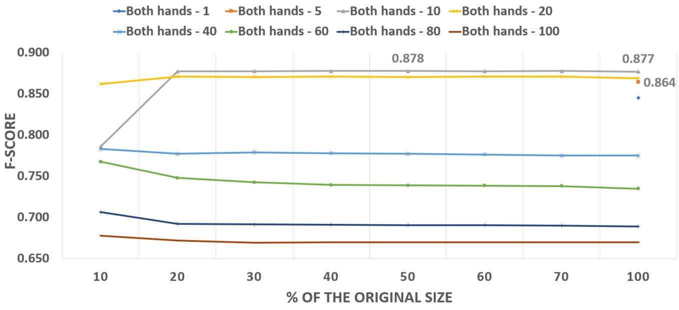

Fig. 6: Average F-score for scenarios with dimensionality reduction for data from video A1 and hands' position

In analyzing the graph in Figure 6, we noticed that the classification models' performance does not suffer significant degradation as the dimensionality reduction increases, with the exception of one case (original window with size 10) in which a drop in performance occurs at the last level of reduction. This scenario points out an advantage in the procedure that carries out windowing and dimensionality reduction over the procedure that only carries out windowing, when original windows with size 10 and 20 are considered (the best windows according to the analysis of Figure 5). As an example, consider the case of the original window with size 10. The size reduction in 50% means that the classification model receives a series with five values as input and achieves an average performance of 0.878. This performance is equivalent, or slightly better, than that obtained for the original window with size 5 (0.864).

Similar behaviors were observed for videos A3, C1 and C3. The differences refer to: *(a - video A3)* for original window with size 10, the classification models reach an average F-score $= 0.822$ when only 20% of the original size is being used (a rate reduction of 80%) against the average F-score $= 0.805$ for the windows in the original size, and against the average F-score $= 0.793$ got with windows whose original size is 5; *(b - video C1)* in this case there was a small gradual decrease in the performance of the classification models. The performance for windows with size 10 decreased from F-score $= 0.705$ in its original size to a F-score $= 0.675$ in a window with 20% of its original size (a rate reduction of 80%). The performance in the window with original size 5 achieved F-score $= 0.694$.

2) *Environment #B:* Figure 7 presents the results for video A1, obtained using the information related to hands velocity. Again, there is a tendency for better performance on smaller windows, and loss of performance on larger windows. The best F-score (0.790) was obtained on window with size 5, using information from both hands. However, the results obtained with both hands were superior only in smaller windows. From the window with size 20, the representation using both hands information is surpassed by the results obtained with the representation using only the right hand information.

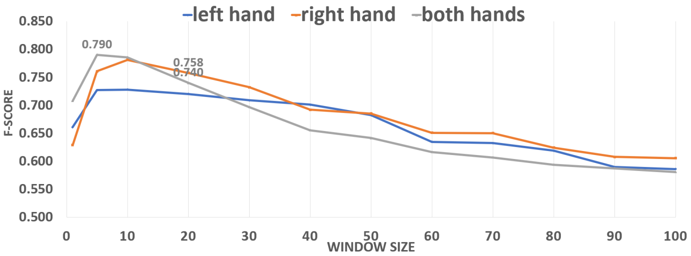

Fig. 7: Average F-score for each window size built with data from video A1 and hands' velocity information

In the analysis of the other videos we observed that: *(a - video A3)* the best F-scores (≈ 0.65) were obtained on windows larger than 30, highlighting the window with size 40 (F-score $= 0.653$) and information only from the right hand; *(b - videos C1, C3)* the best F-scores (≈ 0.57 for video C1; ≈ 0.60 for video C3) were obtained in windows with sizes close to 10 and using the information only from the right hand. The results related to information from both hands were slightly inferior, but they showed the same behavior trend.

The results of the classification models for video A1 after the dimensionality reduction are shown in Figure 8. The results obtained on window with size 10 after the dimensionality reduction in 80% (meaning the use of two values in the series) achieves an F-score $= 0.809$, slightly higher than the F-score achieved with the original window with size 5 (0.786), showing that PAA reduction over windowing can be more informative than the simple windowing. In addition, we can see that the dimensionality reduction led to better F-scores for all cases shown in the figure. This reinforces the advantage of using the reduction in the G-units segmentation problem when the representation based on velocity is applied.

From the results obtained in the other videos, we observed compliance with what was observed in Figure 8, with regard to the best size of windows. In video C3, although the performance improvement occurred in all cases when comparing the result of the original window with the results in the respective reduced versions, there was no gain in relation to the better performance obtained with the original window with size 5.

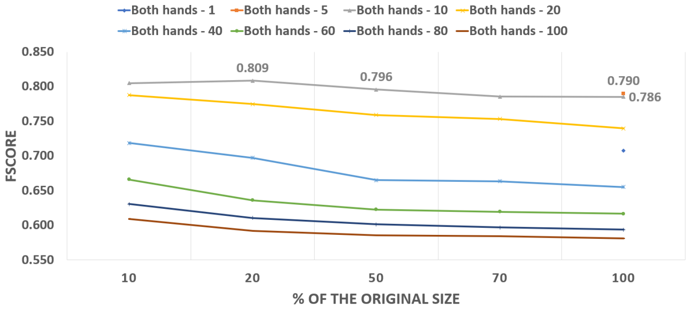

Fig. 8: Average F-score for scenarios with dimensionality reduction for data from video A1 and hands' velocity

C. Experiment #2 - Multiclass classification

1) Environment #A: Figure 9 presents the average accuracy considering all classification models obtained for the same window size using data from video A1. This graph shows that the best results were for windows smaller than 30. The best accuracy (0.718) was obtained in windows with size 20, using information from both hands.

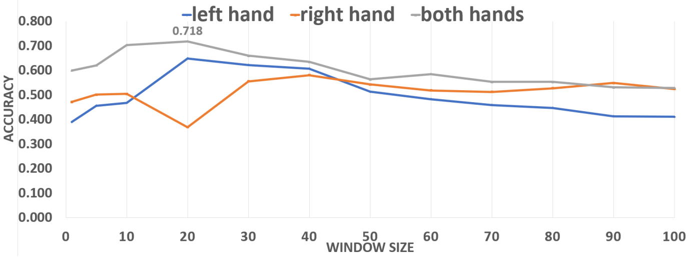

Fig. 9: Average Accuracy for each window size built with data from video A1 and both hands' position information

In terms of accuracy, the results obtained for the other videos were lower: ≈ 0.63 for video A3 in the windows with sizes close to 30; ≈ 0.56 for video C1 in the windows with sizes bigger than 20; and ≈ 0.57 for video C3 in windows with size 20 with performance drop for all other sizes of windows. The use of information from both hands or from the right hand achieves the informed performances.

Figure 10 shows the impact of the dimensionality reduction on the performance of the multiclass classification models, using information from both hands for video A1. We observed again that the performance degradation of the models is very low, motivating the use of dimensionality reduction. The best results were obtained for windows with size 10 and 20. Up to the reduction limit in 80%, such performances suffered slight degradation as the dimensionality is reduced. This test case shows that the reduction in dimensionality leads to a series of only 2 and 4 values that are still capable of maintaining quality information about the gesture phases. Note that in the series originally composed by five values (original window with size 5), the rating model achieved accuracy = 0.620. Such value is much lower than the lowest value (0.681) obtained with reductions over the original windows with size 20.

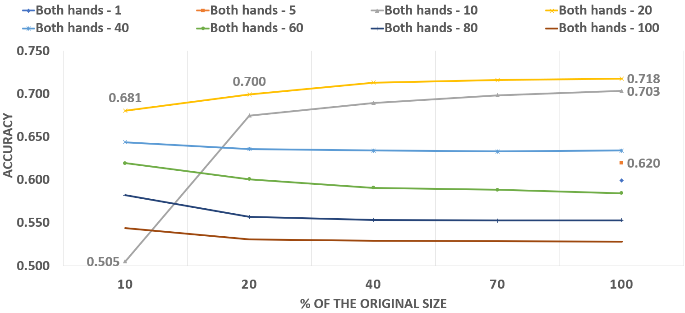

Fig. 10: Average Accuracy for scenarios with dimensionality reduction for data from video A1 and both hands' position

Still analysing the information in Figure 10, we highlight the performance of the classification models in the windows that were originally bigger (bigger than 40). For these windows, there was an increase in the average accuracy as the series were reduced. This fact indicates that the dimensionality reduction is improving the quality of the information, probably reducing the noise level. For the other videos, the observed behavior trends are equivalent to what was described for video A1.

For multiclass classification, we can also analyze the effects of dimensionality reduction considering the special types of errors. Figure 11 shows the behavior of errors while dimensionality is reduced, along with the behavior of accuracy and relaxed accuracy[4], considering a window with size 10, video A1 and position of both hands. The behavior shown in this graph is similar to that observed in the other tests. As a result of this analysis, we can conclude that the drop in performance was characterized by internal errors and, therefore, the reduction in dimensionality impacts accuracy more in terms of this type of error than in terms of errors at the edge.

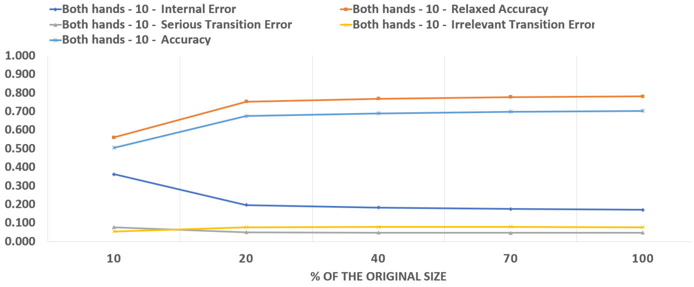

Fig. 11: Special errors analysis

2) Environment #B: Figure 12 shows the average performance of the multiclass classification models, on the A1 video, using representation based on hands velocity. The best results were obtained in windows smaller than 20 and the performance of the classifiers gradually decreases with the increase in window sizes. The highest accuracy (= 0.631) was obtained in windows with size 10, with similar performances with information from both hands or just from the right hand.

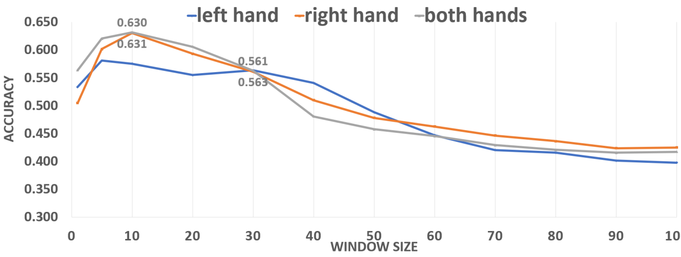

Fig. 12: Average Accuracy for each window size built with data from video A1 and hands' velocity information

In general, for the other videos, the accuracy value achieved by the classification models is lower. For videos A3 and C3,

[4] Here we considered that ITE errors can be seen as correct classifications, then: Relaxed accuracy = accuracy + irrelevant transition errors.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

the best accuracy was obtained in windows with size 20. Only in video C3, the use of information from the right hand surpassed the use of information from both hands.

Figure 13 presents the average accuracy obtained by the multiclass classification models in the scenario of dimensionality reduction for video A1 and representation based on hands velocity. The graph shows that the dimensionality reduction was positive for all windows. For windows with size 10 and 20, the accuracy was 0.631 and 0.606 in the original size, and 0.640 and 0.639 in windows with 90% reduction. In relation to the other videos, we highlight the significant gain that the reduction brought to the classification models obtained for the C1 video in large windows. Figure 14 shows such results.

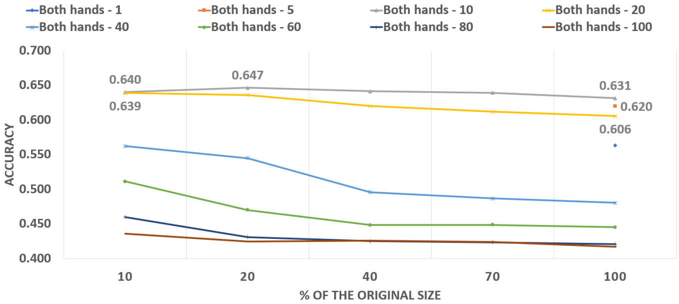

Fig. 13: Average Accuracy for scenarios with dimensionality reduction for data from video A1 and both hands' velocity

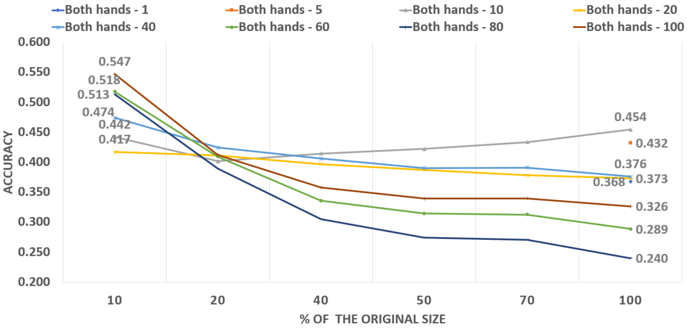

Fig. 14: Average Accuracy for scenarios with dimensionality reduction for data from video C1 and both hands' velocity

VI. Conclusions

In this paper, we presented the results of a systematic study on the effects of dimensionality reduction in the context of the gesture phase segmentation task. The results pointed out that a simple method of dimensionality reduction applied to windowed data can improve the classification models applied to gesture phases segmentation tasks. For binary classification, both in terms of position and velocity based representation, we found that it is possible to reduce the data series represented in a window by up to 80% of the original window size, without significant loss of performance. Slight improvements on performance have been seen for larger original size windows. For multiclass classification with position based representation, we state the classification models performance remained stable with the reduction. However, in all videos, there were cases of drastic drop in performance for small original windows when the reduction was over 80%. Finally, considering the velocity based representation, the dimensionality reduction caused significant improvements in the performance of almost all classification models carried out over larger original size windows. Future works may evolve this study by exploring other classification algorithms to verify whether the impact of the dimensionality reduction with PAA remains as observed herein and brings benefits in the face of the related works.

References

[1] N. Krishnaswamy, P. Narayana, I. Wang, K. Rim, R. Bangar, D. Patil, G. Mulay, R. Beveridge, J. Ruiz, B. Draper, and J. Pustejovsky, "Communicating and acting: Understanding gesture in simulation semantics," in *12th Int. Conf. on Computational Semantics*, 2017.

[2] S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, and J. Malik, "Learning individual styles of conversational gesture," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 3497–3506.

[3] R. C. B. Madeo, S. M. Peres, and C. A. de Moraes Lima, "Gesture phase segmentation using support vector machines," *Expert Systems with Applications*, vol. 56, pp. 100–115, 2016.

[4] D. McNeill, *Hand and Mind: What Gestures Reveal About Thought*. Chicago and London: University of Chicago Press, 1992.

[5] J. J. LaViola Jr., "3D gestural interaction: The state of the field," *ISRN Artificial Intelligence*, vol. 2013, pp. 1–18, 2013.

[6] R. Lun and W. Zhao, "A survey of applications and human motion recognition with microsoft kinect," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 5, pp. 1–48, 2015.

[7] R. C. B. Madeo, C. A. M. Lima, and S. M. Peres, "Studies in automated hand gesture analysis: An overview of functional types and gesture phases," *Lang. Resour. and Evaluat.*, vol. 51, no. 2, pp. 547–579, 2017.

[8] A. Kendon, *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

[9] S. Kita, I. v. Gijn, and H. van der Hulst, "Movement phases in signs and co-speech gestures, and their transcription by human coders," in *Int. Work. on Gest. and Sign Lang. in Hum.-Comp. Inter.*, 1998, pp. 23–35.

[10] C.-S. J. Chu, "Time series segmentation: A sliding window approach," *Information Sciences*, vol. 85, no. 1, pp. 147–173, 1995.

[11] P. K. Wagner, S. M. Peres, R. C. B. Madeo, C. A. M. Lima, and F. de Almeida Freitas, "Gesture unit segmentation using spatial-temporal information and machine learning," in *27th Int. Florida Artificial Intelligence Research Society Conf.*, 2014.

[12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers, 2011.

[13] R. C. B. Madeo, C. A. M. Lima, and S. M. Peres, "A review on temporal reasoning using support vector machines," in *19th Int. Symp. on Temporal Representation and Reasoning*, 2012, pp. 114–121.

[14] R. C. B. Madeo, S. M. Peres, and C. A. M. Lima, "Overview on support vector machines applied to temporal modeling," in *10th National Meeting of Artificial Intelligence*, 2012.

[15] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[16] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems J.*, vol. 3, no. 3, pp. 263–286, 2001.

[17] R. C. B. Madeo, C. A. M. Lima, and S. M. Peres, "Gesture unit segmentation using support vector machines: Segmenting gestures from rest positions," in *28th ACM Symp. on Appl. Comput.*, 2013, pp. 46–52.

[18] R. Artstein and M. Poesio, "Inter-coder agreement for computational linguistics," *Computat. Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.

[19] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[20] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *8th Work. on Research Issues in Data Mining and Knowl. Discov.*, 2003, p. 2–11.

[21] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in *2nd Work. on Temporal Data Mining*, 2002, pp. 53–68.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE