Gesture Unit Segmentation using Support Vector Machines: Segmenting Gestures from Rest Positions

Renata C. B. Madeo
Universidade de São Paulo
Arlindo Bettio, 1000
São Paulo, Brazil
renata.si@usp.br

Clodoaldo A. M. Lima
Universidade de São Paulo
Arlindo Bettio, 1000
São Paulo, Brazil
c.lima@usp.br

Sarajane M. Peres
Universidade de São Paulo
Arlindo Bettio, 1000
São Paulo, Brazil
sarajane@usp.br

ABSTRACT

Gesture analysis has been widely used for developing new methods of human-computer interaction. The advancement reached in the gesture analysis area is also motivating its application to automate tasks related to discourse analysis, such as the gesture phases segmentation task. In this paper, we present an initiative that aims at segmenting gestures, especially considering the "units" – the larger grain involved in gesture phases segmentation. Thereunto, we have captured the gestures using a Xbox Kinect™ device, modeled the problem as a classification task, and applied Support Vector Machines. Moreover, aiming at taking advantage from the temporal aspects involved in the problem, we have used several types of data pre-processing in order to consider time domain and frequency domain features.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition Applications**]: Computer vision; I.2.7 [**Natural Language Processing**]: Discourse

General Terms

Design, Experimentation

Keywords

Gesture Analysis, Gesture Segmentation, Gesture Unit, Support Vector Machine, Temporal Modeling

1. INTRODUCTION

Recently, there has been an increasing interest in gesture analysis research. Most research in this area focus on developing new methods for human-computer interaction, based on the recognition of a pre-defined set of simple gestures or reduced scopes within sign language. Also, some studies investigate multimodal interaction, combining gesture and speech, for instance.

Moreover, gesture analysis may be also used for developing tools aiming at helping linguists to analyze the interaction between speech, gestures, and discourse. These tools could automate some laborious or time-consuming tasks within their process of gesture analysis. One of these tasks is the segmentation of gesture phases.

According to Kendon [5], a person may make one or several *movement excursions* within a discourse. These excursions refer to moving the hands from some position of rest to some region in the space where the main movement occurs, and then turning back to some position of rest. This entire excursion is called *gesture unit*, while the positions between these excursions are called *rest positions*. A gesture unit may be segmented in gesture phases, that can be: *preparation*, in which the hand moves to the position where the gesture content must be expressed; *pre-stroke hold*, which is a brief pause at the end of preparation phase; *stroke*, which contains the peak of effort of the gesture and express its semantic content; *post-stroke hold*, which is a brief pause at the end of the stroke; and *retraction*, in which the hand returns to the rest position.

In Kita et al [6], it is also suggested that pre-stroke hold, stroke and post-stroke hold compose an expressive phase, which also can be composed by a single *independent hold*, that is, a pause that express the semantic content of the gesture. Finally, both Kita et al [6] and Kendon [5] consider the concept of gesture phrase. However, since there is not a consensus about this concept – Kita et al [6] claim that it corresponds to all phases surrounding a single expressive phase (which would include retraction), while Kendon [5] defines it as a segment containing preparation and expressive phase; and it is usually not used for segmenting gestures, we will not consider the concept of gesture phrase in our analysis. The hierarchy of gesture phases based on Kita et al [6] is illustrated in Figure 1.

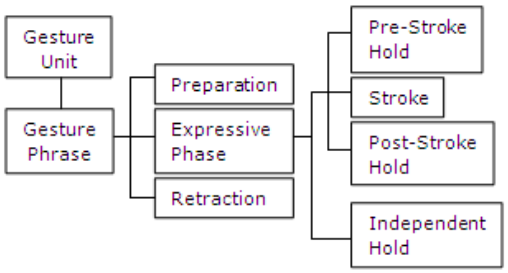

Figure 1: Hierarchy of gesture phases according to Kita et al [6], comprising gesture unit, gesture phrase, and gesture phases.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'13 March 18-22, 2013, Coimbra, Portugal.
Copyright 2013 ACM 978-1-4503-1656-9/13/03 ...$10.00.

The gesture phases segmentation is important for investigating issues such as the synchrony between speech and gestures, analysis related to gesture categorization, or how discourse is produced. Also, it could be used in methods for human-computer interaction, for segmenting the expressive phase that must be analyzed.

In this context, it is necessary to mention that there are difficulties regarding gesture segmentation tasks which come from linguistic area. Some issues about gestures are not well-defined in linguistics researches, such as: if we must analyze the two hands movements as an unique information unit, or if each hand produces distinct information; or how can we interpret movements that occurs in the rest position. Nevertheless, there is some degree of disagreement between gesture segmentations made by two different human coders[1], what can raise important questions about how to evaluate the automated methods, as we will discuss later, in this text.

This paper aims at presenting an initiative for segmenting gesture phases, focusing on the segmentation of gesture units, that is, classifying frames within a video in *rest position* or *gesture*, which represents a first step towards gesture phases segmentation. In order to accomplish this task, we investigate the use of Support Vector Machines (SVM) and the application of several pre-processing methods for extracting time-domain, frequency-domain, and time-frequency domain features, aiming at benefiting from temporal aspects of the problem. The organization of this paper is as follows: some related studies are presented in Section 2; some concepts about SVM are described in Section 3; Section 4 describes our phase gesture segmentation approach, including the dataset used in the experiments; the experiments parameters and the reached results are presented in Section 5; Section 6 presents a comparison between our approaches and related works; and Section 7 presents our final considerations.

2. RELATED WORKS

There are several studies about gesture phase segmentation in linguistic area [5, 6, 11]. Unfortunately, we have not found many studies focusing on the automation of these tasks. However, it is possible to note that there is a recent interest in this problem, since the only two studies that focus on segmenting all gesture phases are from 2007 and 2011. These studies focus on segmenting phases within gesture units: Martell and Kroll [10] consider only gesture units (not explaining how gesture units were segmented), which are analyzed using Hidden Markov Model aiming at defining a phase (preparation, stroke, hold, and retraction) to each frame; Ramakrishnan [13] detects rest position by analyzing frequent positions within the video, obtaining 87% of precision and 4% of false positive rate, and applying SVM to classify if *a priori* detected inflexion frames – i.e., points of transitions between phases – correspond to the beginning of a preparation, stroke, hold, or retraction phase.

Also, there are studies that focus on specific tasks within gesture phase segmentation. For instance, Gebre et al [2] aim at detecting gesture strokes, obtaining an average of 47.24% of precision and 34.41% of recall; and Bryll et al [1] apply a heuristic method to detect hand holds in natural conversation, reaching 82% of precision and 86.4% of recall.

[1]Coder is the name given to the person who segments gesture phases within the discourse in Kita et al [6].

Other studies use some knowledge about gesture phase segmentation to produce more realistic animations. Majkowska et al [9] use Dynamic Time Warping to identify gesture phases, based on their velocity and acceleration profiles, aiming at improving the alignment between hand and body animation. Levine et al [7] also use gesture phases segmentation as an intermediate step in order to improve language animations.

Additionally, Tsai and Lin [15] use a motion history analysis to detect gesture phases in order to provide information for an online tracking technique; Wilson and Bobick [17] use a Finite-State Machine to segment gestures in transition phases (preparation and retraction) and strokes, aiming at differing biphasic gestures – corresponding to gestures with no transition phase – from triphasic gestures – gestures that include transition and stroke phases. In this same work, rest positions are determined heuristically, considering frequent positions within the video, reaching approximately 82% of precision and 79% of recall[2].

Since these last studies [7, 9, 15, 17] do not focus on gesture phase segmentation as their final goal, there are no clear results for phases segmentation. However, they motivate the study of gesture phase segmentation, by presenting some useful applications.

3. SUPPORT VECTOR MACHINES

SVM is based on performing a nonlinear mapping on input vectors from their original feature space into a high-dimensional feature space, and optimizing a hyperplane capable of separating data in this high-dimensional feature space [16].

Considering a training set with N samples, defined by $\{\boldsymbol{x_i}, y_i\}_{i=1}^{N}$, where $\boldsymbol{x_i}$ is an input, y_i is an output, and $y_i \in \{-1, +1\}$, the goal of SVM is finding an optimal classes separation hyperplane, which is given by $h(\boldsymbol{x_i}) = \boldsymbol{w}^T\varphi(\boldsymbol{x_i}) + b$, where $\boldsymbol{w}$ is the optimal set of weights, b is the optimal bias, and φ is the nonlinear mapping applied to input vectors. SVM optimizes the hyperplane maximizing the distance between this hyperplane and its closest datapoints ($\boldsymbol{x_i}$), which corresponds to minimizing $\boldsymbol{w}$ using, for example[3]:

$$\min\ \phi(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

where C is a regularization factor, and ξ is an error factor, subject to

$$y_i(\boldsymbol{w}^T\varphi(\boldsymbol{x_i}) + b) \geq 1 - \xi, \quad i = 1, ..., N,$$
$$\xi_i \geq 0, \quad i = 1, \cdots, N.$$

Applying Lagrangian method in Eq. (1), we obtain

$$\max\ \mathcal{L}_1(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \langle\varphi(\boldsymbol{x_i}) \cdot \varphi(\boldsymbol{x_j})\rangle, \tag{2}$$

subject to

[2]Values for precision and recall were not provided explicitly in Wilson and Bobick [17]. These values were estimated by the authors of the present paper, through the analysis of the figure comparing the human-labeled video with the automatic labeled video.

[3]Considering a soft margin optimization using a 1-norm.

$$\sum_{i=1}^{N} \alpha_i y_i = 0,$$
$$C \geq \alpha_i \geq 0 \quad \text{for } i = 1, 2, \cdots, N,$$

where $\boldsymbol{\alpha}$ are Lagrangian multipliers. Solving the problem in Eq. (2), it is possible to solve the problem in Eq. (1), since $\boldsymbol{w}$ can be defined in terms of $\boldsymbol{\alpha}$ [3]. In Eq. (2), a kernel function $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ can be used to represent the dot product $\langle \varphi(\boldsymbol{x_i}) \cdot \varphi(\boldsymbol{x_j}) \rangle$, performing an implicit nonlinear mapping. In this paper, we consider Radial Basis Function (RBF) as kernel function, which is given by

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(\frac{\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2}{2\delta^2}\right), \tag{3}$$

where δ is the RBF parameter.

4. PHASES GESTURE SEGMENTATION

In this section, we present a definition of gesture phases segmentation problem and a description of the dataset used in the experiments.

4.1 Problem Definition

In this work, a video represented by a sequence of frames $S = \{f_1, f_2, \cdots, f_{nf}\}$ is input to a segmentation strategy aiming at identifying gesture phases. The segmentation problem consists in receiving the representation of a frame f_i as input and classifying these frames as one class among $c_i = \{E, P, S, H, R\}$, corresponding to **r**est, **p**reparation, **s**troke, **h**old, and **r**etraction.

We have divided this classification problem into smaller subproblems:

1. Classifying rest positions: input $f_i \in S$, and output $c_i = \{E, G\}$, where $G \supset \{P, S, H, R\}$ corresponding to **g**esture units. This is the main focus of this paper.
2. Classifying holds: input $f_i \in S_G$, where $S_G \supset G$, and output $c_i = \{H, D\}$, where $D \supset \{P, S, R\}$ corresponding to **d**ynamic phases.
3. Classifying strokes: input $f_i \in S_M$, where $S_M \supset M$, and output $c_i = \{S, T\}$, where $T \supset \{P, R\}$ corresponding to **t**ransition phases.
4. Classifying preparation and retraction: input $f_i \in S_T$, where $S_T \supset T$, and output $c_i = \{P, R\}$.

Figure 2 illustrate our strategy for dividing the classification problem into subproblems.

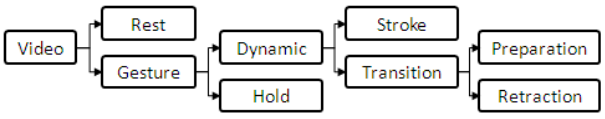

Figure 2: Strategy for classifying gesture phases.

4.2 Dataset and Data Representation

The data consists in streams of gestures made by a person telling a story, captured by an application based on Microsoft Xbox Kinect™. The storytelling is motivated by comics, shown to the storyteller before the capture session. As the comics theme is commonly known by the storytellers, the storytelling is an easy task.

In this case, the Xbox Kinect™ device is used to capture the RGB frames and, for each frame, the 3D positions of six points of interest in the storyteller body (hands, wrists, head, and spine). Our dataset consists in two different videos of the same person telling two different stories. Table 1[4] presents the information about the two videos. There are 3012 frames, and each frame of these videos was labeled by one human coder, in order to allow building and validating our segmentation approach.

Video	Length	Frames#	Rest Position#	Gesture Unit#			
1	60s	1747	698	1049			
				P	S	R	H
				146	656	208	39
2	40s	1264	493	771			
				P	S	R	H
				180	431	106	54

#: number of frames: in the whole video; corresponding to rest position and gesture unit; and corresponding to each phase.

Table 1: Information about the videos used for training and testing.

In order to represent the videos in an adequate way to be processed by the classification algorithm, the information about 3D position was used to create a normalized vector representation: for each frame, the position of hands and wrists is subtracted from the position of the spine, and this new 3-dimensional position is divided by the distance between head and spine.

From the normalized vector representation, new information was created by estimating velocity and acceleration measures. For the velocity, the estimation is given by

$$v_{i,i-d} = \frac{\Delta r_{i,i-d}}{t_i - t_{i-d}},$$

where t is the timestamp of frame i, d is the displacement in frames, and $\Delta r_{(i,i-d)}$ is the Euclidean distance between normalized 3D position of the interest point at frame i and at frame $i - d$. The acceleration is estimated through

$$a_{i,i-1} = \frac{v_i - v_{i-1}}{t_i - (t_{i-1})}.$$

Also, the proposed approach considers a windowed strategy, using information from past and/or future frames to represent each frame of interest due to the intrinsic temporal aspects of gesture phases segmentation problem. Figure 3 illustrates the windowing technique; each cell of the window contains the left hand (lh) and right hand (rh) velocity or acceleration information, depending on what is the considered measure.

5. EXPERIMENTS AND RESULTS

The experiments have been carried out in order to verify the performance of SVM model in the gesture phase segmentation problem, modeled as a classification task. All experiments were performed using MATLAB® with the toolbox

[4]Some frames were not considered because they represented changes in rest position. Interpreting these changes is still an open research problem within Gesture Studies.

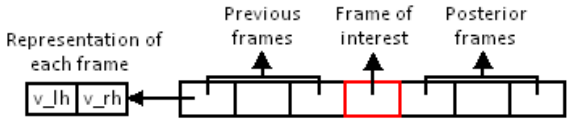

Figure 3: Example of window with 7 frames and fourth frame as frame of interest.

Spider[5]. We have applied a SVM algorithm available in the toolbox, using $C = 100$ and RBF kernel function (see Eq. 3), considering kernel parameters (δ) from 2^{-6} to 2^{4}, varying by one the exponent of the powers of two[6]. In order to accomplish the requirements of training and test of the classification model, we have used a balanced version of video #1 for training each binary problem, and the entire video #2, unbalanced, for testing. The datapoints were obtained using the windowing procedure with 1 to 80 frames[7], centered in the frame of interest.

5.1 Feature Representation Analysis

Firstly, different tests were executed aiming at finding the best parameters to represent the gestures for the segmentation tasks. These parameters are: point of interest (hand and/or wrist); position of the frame of interest; time displacement for calculating velocity; and measured feature (velocity and/or acceleration). This last parameter (measured feature) is evaluated in all these initial tests. Other tests were run, but the results presented here fixes best parameters found in order to analyze each parameter individually.

Points of Interest Analysis

In this work, we have considered four main points of interest to represent the spacial displacement that characterize the gesture: left and right hands, and left and right wrists. However, in this work, left and right hands are considered together, as "hands", as well as left and right wrists are considered together, as "wrists". In order to evaluate the representation power of each pair of points of interest, we have carried out experiments to compare them. Table 2 presents classification errors reached in such experiments[8], where it is possible to observe the superior representatives power when only the hands are used. In these tests, we have fixed $d = 3$, using past frames only, and frame of interest in the middle of the window.

Position of the Frame of Interest Analysis

[5]An object-oriented environment for machine learning in MATLAB®, available at http://people.kyb.tuebingen.mpg.de/spider/main.html.

[6]Whenever the best result for a test was obtained with $\delta = 2^{-6}$, new tests were run with δ from 2^{-8} to 2^{-7} in order to evaluate if smaller values of δ were needed.

[7]Similar to the choice of δ, whenever a test reached its best results with a window size close to 80, new tests were run considering window size until 120, 150, or 180 frames, in order to verify if larger windows were needed.

[8]The first number in parentheses, for all tables in Section 5, refers to the window size for which the smaller error was reached. The second, expressed in powers of two in order to differ from window size, refers to the RBF parameter used to obtain the best SVM model.

	Velocity	Acceleration	Both
Hands	**10.5 (46, 2^{-1})**	**16.2 (28, 2^{-2})**	**12.9 (40, 2^{0})**
Wrists	10.8 (52, 2^{-1})	18.2 (46, 2^{-1})	15.7 (73, 2^{1})
Both	10.7 (79, 2^{0})	16.3 (34, 2^{-1})	13.5 (34, 2^{0})

Table 2: Classification errors (in %) considering the two main points of interest: hands, wrists, or both.

Gesture phases represent the temporal structure of gesture. Thus, gesture phase segmentation is intrinsically a temporal task. From this, we have supposed that it was necessary to analyze information of neighboring frames, justifying our windowed procedure to compose our datapoints. Our initial hypothesis was that it is important to analyze previous and posterior frames. However, we have also tested considering only previous frames. Results shown in Table 3 corroborate our initial hypothesis. In these tests, we have fixed $d = 3$, using past frames only, with hands as point of interest.

	Classification Error	
Position	Velocity	Acceleration
Middle	**10.5 (46, 2^{-1})**	16.2 (28 , 2^{-2})
End	17.8 (19 , 2^{-3})	23.1 (3 , 2^{-5})

Table 3: Classification errors (in %) using different positions for the frame of interest.

Time Displacement Analysis

As mentioned in Section 4.2, a parameter d referring to the time displacement is considered for estimating velocity information. Ramakrishnan [13] estimates velocity for a frame t by considering the displacement between frame $t-5$ and $t+5$ (i.e. $d = 5$), arguing that this practice would avoid the noise produced by calculating velocities over consecutive frames.

In our work, we have evaluated different values for d, considering two approaches: using past frames only ($t - d$); or using past and future frames ($t+d$ and $t-d$). Since the difference between best results for both approaches was not significant, we have decided to consider only past frames since this approach requires a smaller window. Table 4 presents classification errors with $d = 1, 3, 5$ and 7, considering both approaches. In these tests, we have fixed hands as point of interest, with the frame of interest in the middle of the window. From these results, we have chosen to use $d = 3$.

	Velocity		Acceleration	
d	Only past	Past and future	Only past	Past and future
1	12.9 (38, 2^{-2})	12.6 (51, 2^{-1})	17.3 (48, 2^{0})	19.8 (29, 2^{-1})
3	**10.5 (46, 2^{-1})**	**10.2 (80, 2^{-1})**	16.2 (28, 2^{-2})	13.9 (38, 2^{-2})
5	11.1 (58, 2^{-1})	12.7 (16, 2^{-5})	16.9 (30, 2^{-2})	**13.0 (26, 2^{-3})**
7	12.3 (50, 2^{-1})	11.8 (20, 2^{-5})	**12.4 (33, 2^{-2})**	13.6 (28, 2^{-3})

Table 4: Classification errors (in %) considering different values for d, only past frames or past and future frames.

Time-Domain Features Analysis

In our initial windowed approach, all features from frames

within the window are put together into a vector representation. However, it is also possible to extract time-domain features from the window in order to represent each frame. The time-domain features used in this work are described for eletromiogram signals in Phinyomark et al [12], and are briefly described in Table 5. As it can be seen in this table, extracting these time-domain features have not improved the performance obtained by using all features in a vector representation.

Time-Domain Feature Descriptions	Error
Sum of all components of the signal	13.5 (31, 2^{1})
Mean of all components of the signal (M)	13.5 (30, 2^{1})
M using a weighting window function	18.4 (19, 2^{2})
M using a continuous weighting window function	20.6 (10, 2^{0})
Sum of squared components of signal (S)	15.0 (25, 2^{1})
Squared root of S	14.2 (25, 2^{1})
Cumulative length of the waveform over the time segment	25.4 (3, 2^{-7})

Table 5: Classification errors (in %) using different time-domain feature extraction approaches.

Frequency Domain Features Analysis

We have applied an one-dimensional Discrete Fourier Transform (DFT) for each signal (considering left hand and right hand signals separately) and a bi-dimensional DFT (considering left and right hand as two dimensions of the same signal), obtaining amplitudes for each ranges of frequencies. We have used these sequences of amplitudes as input for SVM model, obtaining **13.93%** of error with a window of 26 frames for one-dimensional DFT (RBF parameter: 2^{-1}), and **12.37%** of error with a window of 32 frames for bi-dimensional DFT (RBF parameter: 2^{-1}).

Time-Frequency Domain Features Analysis

For extracting time-frequency domain features, we have used Discrete Wavelet Transform with Daubechies 4 (db4), considering decomposition levels from 1 to 3, using only components D1, D2, and D3, respectively. From these components, we have calculated mean and standard deviation (SD) in order to represent data, as in Lima and Coelho [8]. Table 6 presents the results using such features. Similarly to time-domain feature results, time-frequency domain features have not improved the performance of the classifier model which considers the entire window as input.

Features	D1	D2	D3
Mean	15.3 (23, 2^{-2})	17.5 (23, 2^{-2})	19.1 (23, 2^{-1})
SD	17.4 (28, 2^{-2})	18.1 (26, 2^{-2})	19.3 (30, 2^{-2})
Mean and SD	14.1 (39, 2^{-6})	15.3 (26, 2^{-6})	15.8 (15, 2^{-7})

Table 6: Classification errors (in %) using Discrete Wavelet Transform.

5.2 Final Classifier Analysis

From these results, we have chosen to make a deeper analysis of the classifier model build with windowed data containing frames represented by hands velocity, with frame in the middle of the window as frame of interest. In the sequence of this section, we present some figures to illustrate the conclusions that this final experiment allowed us to reach.

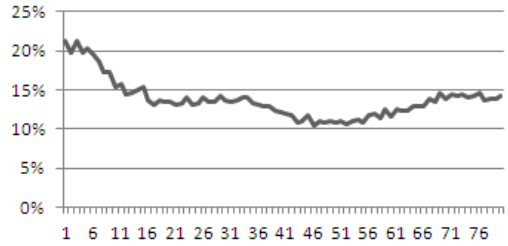

Figure 4: Classification error for classifier models varying the sizes of window.

First, Figure 4 shows that temporal aspects impact performance indeed, since classification error is bigger with no window (i.e., an "one frame window") or small windows. The best performance is achieved with 46 frames in the window.

Second, Figure 5 illustrates the Receiver Operating Curve (ROC), which relates false positive rate (x axis) and recall (y axis), for some models with windows of 40 to 58 frames[9]. As we can see, the window with 46 frames also presents a good performance, presenting 89.32% of recall and 10.44% of false positives.

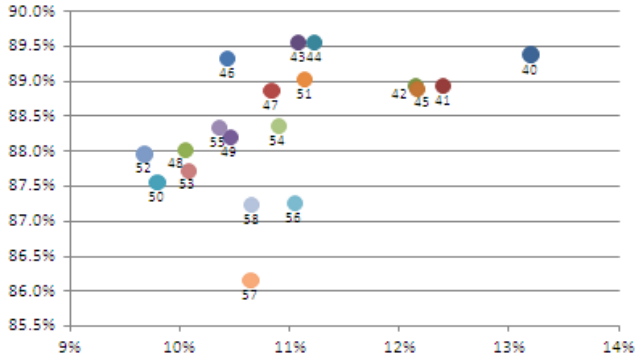

Figure 5: ROC graph for some classifiers models with fixed parameters and with windows of 40 to 58 frames.

Also, we have run a holdout test with this configuration 10 times in order to verify the stability of our approach. We have obtained an average error of 11.09%, with standard deviation of 0.33%, indicating that our approach is indeed stable.

From a linguist point of view, it can be interesting to analyze the nature of errors, in order to better interpret the automated segmentation results. Still considering our best classification model results, from 128 incorrectly classified frames:

- 12 frames (9.38%) compose two really small segments of rest position, with 8 frames and 4 frames;
- 76 frames (59.38%) consist of consecutive errors in the beginning or in the end of the segment, i.e., in the transition between phases. Figure 6 shows an example of transition error;

[9]This range was chosen for containing the greater number of consecutive results in the lower quartile considering classification errors.

- 40 frames (31.25%) are internal errors (incorrect classification that do not belong to the transition between segments), from which 38 frames correspond (or are close) to hold phases – phases with velocity profile similar to rest positions; while the remaining 2 frames (1.56%) had no apparent difficulty in analyzing.

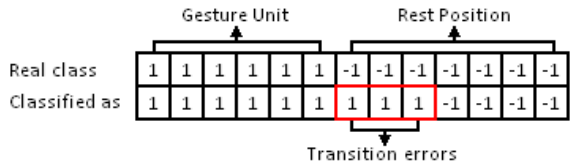

Figure 6: Example illustrating transition errors.

Finally, instead of analyzing errors for each frame, it is possible to analyze the errors for each segment of rest position or gesture unit. In this approach, a segment is considered wrong when it has more than 20% wrong frames as internal errors; or more than 40% wrong frames within all segment. Consecutive incorrectly classified frames in the transition of the frame are considered deviations. Table 7 presents errors for each segment. It is important to highlight that, from three wrong segments of rest position, two correspond to really small segments, with 4 and 8 frames.

Class	Total Segments	Wrong classified segments	Average start deviation (in frames)	Average end deviation (in frames)
Rest position	9	3	4.5	1.1
Gesture unit	9	1	4.1	0.1

Table 7: Errors considering segments of each class.

5.3 Preliminary Results for Gesture Phase Segmentation

We have also performed some preliminary experiments for segmenting hold, stroke, and preparation phases. Our strategy, as mentioned on Section 4.1, consists in binary classifiers. In Table 8, initial results reached for other classifiers models are shown, all built with the same parameters applied in the best classifier to gesture unit segmentation.

Study	# Frames	σ	Precision	Recall	F-score
Hold	6	0.016	75.6	57.4	65.3
Stroke	65	0.25	70.2	88.6	78.4
Preparation	88	2	96.0	83.6	89.4
Retraction	88	2	67.8	90.8	77.6

Table 8: Preliminary results for phase gesture segmentation.

Also, as we have balanced our training sets, there were few examples of hold and non-hold to train our classifier. This practice may explain the low F-score for hold segmentation. This result may be improved by using other strategies to balance classes, or by expanding our dataset.

6. COMPARISON WITH RELATED WORKS

In Section 2, we present some related studies. Most of them do not present results for gesture unit segmentation. However, Ramakrishnan [13] presents some results for identifying rest position, and it is possible to deduct the results for Wilson and Bobick [17]. Table 9 compares our results with these studies, considering the precision, recall and F-score measures.

Study	Precision	Recall	F-score
Our approach	84.3	89.3	86.7
Ramakrishnan [13]	87	96	91.3
Wilson and Bobick [17]	82	79	80.5

Table 9: Comparison between our approach and related studies for gesture unit segmentation.

It is important to highlight that: (a) the approaches in Ramakrishnan [13] and Wilson and Bobick [17] rely on finding frequent hand positions within the video, while ours intend to find a velocity pattern for rest position; (b) all studies use different videos, so it is not possible to evaluate the degree of difficulty for each study.

Concerning to other gesture phases, it is more difficult to compare our preliminary results with previous results in literature, since studies use different strategies.

Firtly, it is important to highlight that, in our work, we use a strategy in order to obtain transition frames, i.e., frames containing preparation and retraction phases, and then we identify preparation and retraction among these transition frames. That means that we have one result for separating preparation and retraction, differently from other studies which segment both preparation and retraction frames from the entire gesture unit.

It is the case of Martell and Kroll [10], that identifies all phases within the gesture unit. This study presents a low F-score for hold detection (36%), and F-scores of 54%, 59%, and 67% for preparation, stroke, and retraction, respectively. As they identified that hold phase introduces many errors, their next tests consider a segmentation in preparation, stroke, and retraction phases only. In this case, their best method presents a F-score of 56%, 68%, and 79% for preparation, stroke, and retraction, respectively.

Ramakrishnan [13] also presents a method for segmenting preparation, stroke, retraction, and hold within a gesture unit. However, Ramakrishnan [13] applies a strategy in which points corresponding to a possible transition between phases are detected and then classified as the beginning of a specific phase. Therefore, metrics for evaluating results rely on correctly labeled segments and average deviations in the beginning or in the end of each segment. Thus, if we disregard deviations and consider errors for segments (and not for frames, as our method does), the method of Ramakrishnan [13] would obtain F-scores of 95.3%, 79.9%, 74.3%, and 85.7%, for hold, stroke, preparation, and retraction, respectively. Considering a similar strategy for evaluating our results (described in Section 5.2), we obtain F-scores of 75%, 64.7%, and 85.7% for identifying holds, strokes, and preparation, respectively.

Also, Gebre et al [2] identifies strokes within a video, reaching a low F-score (38.71%), and Bryll et al [1] identifies holds within a video, reaching a F-score of 84.14%.

7. CONCLUSIONS

This paper has presented a strategy for gesture phase recognition using SVM, focusing on solving gesture unit segmentation problem, which consists in segmenting rest posi-

tion from gesture units within a recorded discourse. Our strategy divides the problem into smaller binary problems, starting from identifying gesture units, and then advancing to identifying gesture phases (holds, strokes, preparation and retraction).

In this work, we have investigated gesture unit segmentation problem through several tests, aiming at finding the best parameters for a SVM classifier in order to distinguish rest position from gesture unit. The investigated parameters were: point of interest; position of the frame of interest; time displacement for calculating velocity; and measured feature. Also, we have explored time domain features, frequency domain features, and frequency-time domain features aiming at improving results for our first model. However, the best result was achieved by using a SVM classifier trained with: a simple windowed datapoint; window with 46 frames; hands as point of interest; velocity as measured feature, considering a time displacement of 3 past frames; and considering the 23rd frame as the frame of interest for classification.

Another important aspect to consider is the level of disagreement of human coders in the same task. Two coders have labeled video #1 in order to evaluate this level of disagreement. In video #1, coders have disagreed in 4.01% of the frames for gesture unit segmentation task. This disagreement corresponds to transition errors, and to the identification of really small rest position segments. Actually, for the latter case, one coder has identified the segment as rest position and the other coder has identified the same segment as gesture unit. These two kind of errors summed up to 68.8% of all errors of our model, i.e., 7.2% of all frames. That is, only 3.29% of all frames present errors which are not common in the task made by human coders.

Although it is not possible to directly compare the results – due to the use of different videos in each analysis, different metrics for evaluation [13], and to different specification of the problems [1]; the comparison of our results with related works shows that our approach is promising.

For gesture unit segmentation, although Ramakrishnan [13] presents better results, his approach is based on a heuristic which consists on identifying frequent hand positions within the video. Thus, his approach may depend on the behavior of the speaker. Our approach is based on velocity profiles and, since rest position generally consists in segments of the video where the hand presents little or no movement, velocity profiles seem to be a more reliable feature for identifying rest positions.

For gesture phases segmentation, our preliminary tests present interesting results, which are further explored in order to obtain more effective conclusions.

The next steps in our research include: (a) validating the results considering other human coders; (b) evaluate all parameters for each gesture phase segmentation problem; and (c) apply other SVM methods that consider temporal reasoning within SVM model, such as SVM with recursive kernels [4] and Support Vector Echo-State Machines [14].

8. ACKNOWLEDGMENTS

The first author thanks São Paulo Research Foundation (FAPESP/Brazil) - process number 2011/04608-8.

9. REFERENCES

[1] R. Bryll, F. Quek, and A. Esposito. Automatic Hand Hold Detection in Natural Conversation. In *IEEE Workshop on Cues in Communication, Kauai, Hawaii*, pages 1–6, 2001.

[2] B. G. Gebre, P. Wittenburg, and P. Lenkiewicz. Towards automatic gesture stroke detection. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, Istanbul, Turkey, May 2012.

[3] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[4] M. Hermans and B. Schrauwen. Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation*, 24:104–133, jan. 2012.

[5] A. Kendon. *Gesture: Visible Action as Utterance*. Cambridge University Press, 2005.

[6] S. Kita, I. van Gijn, and H. van der Hulst. Movement phases in signs and co-speech gestures, and their transcription by human coders. In I. Wachsmuth and M. Frohlich, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume 1371 of *Lecture Notes in Computer Science*, pages 23–35. Springer Berlin / Heidelberg, 1998.

[7] S. Levine, C. Theobalt, and V. Koltun. Real-time prosody-driven synthesis of body language. *ACM Trans. Graph.*, 28(5):172:1–172:10, Dec. 2009.

[8] C. A. M. Lima and A. L. V. Coelho. Kernel machines for epilepsy diagnosis via eeg signal classification: A comparative study. *Artif. Intell. Med.*, 53(2):83–95, Oct. 2011.

[9] A. Majkowska, V. B. Zordan, and P. Faloutsos. Automatic splicing for hand and body animations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, SCA '06, pages 309–316, 2006.

[10] C. H. Martell and J. Kroll. Corpus-based gesture analysis: an extension of the form dataset for the automatic detection of phases in a gesture. *International Journal of Semantic Computing*, 1:521–536, 2007.

[11] D. McNeill. *Gesture and Thought*. University of Chicago Press, 2005.

[12] A. Phinyomark, C. Limsakul, and P. Phukpattaranont. A novel feature extraction for robust emg pattern recognition. *Journal of Computing*, 1:71–80, 2009.

[13] A. S. Ramakrishnan. Segmentation of hand gestures using motion capture data. Master's thesis, University of California, 2011.

[14] Z. Shi and M. Han. Support vector echo-state machine for chaotic time-series prediction. *IEEE Trans. on Neural Networks*, 18(2):359 –372, mar. 2007.

[15] T.-H. Tsai and C.-Y. Lin. Visual hand gesture segmentation using three-phase model tracking technique for real-time gesture interpretation system. *Journal of Information Hiding and Multimedia Signal Processing*, 3(2):122–134, Apr. 2012.

[16] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[17] A. Wilson, A. Bobick, and J. Cassell. Recovering the temporal structure of natural gesture. In *Automatic Face and Gesture Recognition, 1996., Proc. of the Second International Conference on*, pages 66 –71, oct 1996.