Simultaneous segmentation and recognition of gestures for human-machine interaction

Harold Vasquez, L. Enrique Sucar, Hugo Jair Escalante
Department of Computational Sciences
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Tonantzintla, 72840, Puebla, Mexico.
{hvasquez,esucar,hugojair}@inaoep.mx

Abstract

Human-activity and gesture recognition are two problems lying at the core of human-centric and ubiquitous systems: knowing what activities/gestures users are performing allows systems to execute actions accordingly. State-of-the-art technology from computer vision and machine intelligence allow us to recognize gestures at acceptable rates when gestures are segmented (i.e., each video contains a single gesture). In ubiquitous environments, however, continuous video is available and thus systems must be capable of detecting when a gesture is being performed and recognizing it. This paper describes a new method for the simultaneous segmentation and recognition of gestures from continuous videos. A multi-window approach is proposed in which predictions of several recognition models are combined; where each model is evaluated using a different segment of the continuous video. The proposed method is evaluated in the problem of recognition of gestures to command a robot. Preliminary results show the proposed method is very effective for recognizing the considered gestures when they are correctly segmented; although there is still room for improvement in terms of its segmentation capabilities. The proposed method is highly efficient and does not require learning a model for *no-gesture*, as opposed to related methods.

1 Introduction

Human-computer interaction technology plays a key role in ubiquitous data mining (i.e., the extraction of interesting patterns from data generated in human-centric environments), see [Eunju, 2010]. From all of the alternative forms of interaction, gestures are among the most natural and intuitive for users. In fact, gestures are widely used to complement verbal communication between humans. Research advances in computer vision and machine learning have lead to the development of gesture recognition technology that is able to recognize gestures at very acceptable rates [Aggarwal and Ryoo, 2011; Mitra, 2007]. However, most of the available methods for gesture recognition require gestures to be segmented before the recognition process begins [Aviles *et al.*, 2011]. Clearly, this type of methods is not well suited for ubiquitous systems (and real applications in general), where the recognition of gestures must be done from a continuous video in real time [Eunju, 2010; Huynh *et al.*, 2008].

This paper introduces a new approach for the simultaneous segmentation and recognition of gestures in continuous video. The proposed method implements a voting strategy using the predictions obtained from multiple gesture models evaluated at different time-windows, see Figure 1. Windows are dynamically created by incrementally scanning the continuous video. When the votes from the multiple models favor a particular gesture, we segment the video and make a prediction: we predict the gesture corresponding to the model that obtained the majority of votes across windows.

We use as features the body-part positions obtained by a Kinect^{TM} sensor. As predictive model we used Hidden Markov Models (HMMs), one of the most used for gesture recognition [Aviles *et al.*, 2011; Aggarwal and Ryoo, 2011; Mitra, 2007]. The proposed method is evaluated in the problem of recognition of gestures to command a robot. Preliminary results show the proposed method is very effective for recognizing the considered gestures when they are correctly segmented. However, there is still room for improvement in terms of its segmentation capabilities. The proposed method is highly efficient and does not require learning a model for *no-gesture*, as opposed in related works.

The rest of this paper is organized as follows. The next section briefly reviews related works on gesture spotting. Section 3 describes the proposed approach. Section 4 reports experimental results that show evidence of the performance of proposed technique. Section 5 outlines preliminary conclusions and discusses future work direction.

2 Related work

Several methods for the simultaneous segmentation and recognition of gestures (a task also known as gesture spotting) have been proposed so far [Derpanis *et al.*, 2010; Yuan *et al.*, 2009; Malgireddy *et al.*, 2012; Kim *et al.*, 2007; Yang *et al.*, 2007]. Some methods work directly with spatiotemporal patterns extracted from video [Derpanis *et al.*, 2010; Yuan *et al.*, 2009]. Although being effective, these methods are very sensitive to to changes in illumination, scale, appearance and viewpoint.

On the other hand, there are model-based techniques that use the position of body-parts to train probabilistic models (e.g., HMMs) [Aggarwal and Ryoo, 2011; Mitra, 2007]. In the past, these type of methods were limited because of the need of specialized sensors to obtain body-part positions. Nowadays, the availability of Kinect^{TM} (which can extract skeleton information in real time) has partially circumvented such limitation [Webb and Ashley, 2012].

Besides the data acquisition process, some of these methods require the construction of a no-gesture model (e.g., [Kim *et al.*, 2007]) or transition-gesture model (e.g., [Yang *et al.*, 2007]). The goal of such models is to determine within a video when the user (if any) is not performing any gesture or the transition between different gestures. Building a model for *no-gesture* is a complicated and subjective task that depends on the particular application where the gesture recognition system is to be implemented [Kim *et al.*, 2007]. In ubiquitous systems, however, we want gesture recognition methods to work in very general conditions and under highly dynamic environments. Hence, a model for *no-gesture* is much more complicated to generate in these conditions.

Finally, it is worth to mention that many of the available techniques for gesture spotting can be very complex to implement. This is a particularly important aspect to consider for some domains, for example in mobile devices and/or for human-robot interaction; where there are limited resources and restricted programming tools for the implementation of algorithms. Thus, sometimes simplicity is preferred at the expense of loosing a little bit in precision in these domains.

The method we propose in this paper performs segmentation and recognition of gestures simultaneously and attempts to address the limitations of most of the available techniques. Specifically, our proposal is efficient and very simple to implement; it is robust, to some extend, to problems present in appearance-based methods; and, more importantly, does not require the specification of a *no-gesture* model.

3 Multiple-windows approach

We face the problem of simultaneously segmenting and recognizing gestures in continuous video[1]. That is, given a sequence of images (video) we want to determine where a gesture is being performed (independently of the type of gesture) and next to recognize what is the actual gesture being performed. We propose a solution based on multiple windows that are incrementally and dynamically created. Each window is passed through predictive models each trained to recognize a particular gesture. The predictions of models for different windows are accumulated, when the model for a particular gesture obtains a majority of votes, we segment the video and make a prediction, cf. Figure 1.

The underlying hypothesis of our work is that when a window covers a large portion of a particular gesture, the confidence in the prediction of the correct model will be high and those of other models will be low. Accumulating predictions allow us to be more confident in that the gesture is being performed within a neighborhood of temporal windows.

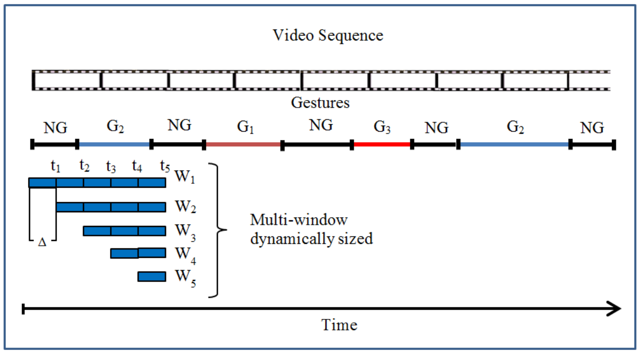

Figure 1: Graphical illustration of the proposed approach. On the top we show a video sequence that can be divided into sections of no gesture (NG) and gesture, which are identified by the class of gesture (G_1, G_2, G_3). Below we illustrate a series of windows that are dynamically created and extended each Δ time units. That is, at the beginning W_1 is created, then at t_1, W_2 is created and W_1 is extended by Δ, and so on. At t_5 there are 5 windows of different size, for each window we estimate the probability of all gestures using HMMs.

The rest of this section describes in detail the proposed technique. First we describe the considered features, next the predictive models and finally the approach to simultaneous segmentation and recognition of gestures.

3.1 Features

We use the information obtained through a Kinect^{TM} as inputs for our gesture spotting method. The Kinect^{TM} is capable of capturing RGB and depth video, as well as the positions of certain body-parts at rates up to 30 frames-per-second (fps). In this work we considered gestures to command a robot that are performed with the hands. Therefore, we used the position of hands as given by Kinect^{TM} as features. For each hand, we obtain per each frame a sextuple indicating the position of both hands in the *x, y,* and *z* coordinates. Since we consider standard hidden Markov models (HMMs) for classification, we had to preprocess the continuous data provided by the considered sensor. Our preprocessing consisted in estimating tendencies: we obtain the difference in the positions obtained in consecutive frames and codify them into two values: $+1$ when the difference is positive and a 0 when the difference is zero or negative. Thus, the observations are sextuples of zeros and ones (the number of different observations is 2^6). These are the inputs for the HMMs.

3.2 Gesture recognition models

As classification model we consider an HMM[2], one of the most popular models for gesture recognition [Aviles *et al.*, 2011; Aggarwal and Ryoo, 2011; Mitra, 2007]. For each gesture i to be recognized we trained an HMM, let $\mathcal{M}_i$ denote the

[1] Although we use (processed) body-part positions as features, we refer to the sequence of these features as video. This is in order to simplify explanations.

[2] We used the HMM implementation from Matlab^{R}'s statistics toolbox.

HMM for the i^{th} gesture, where $i = \{1, \ldots K\}$ when considering K different gestures. The models are trained with the Baum-Welch algorithm using complete sequences depicting (only) the gestures of interest. Each HMM was trained for a maximum of 200 iterations and a tolerance of 0.00001 (the training process stops when changes between probabilities of successive transition/emission matrices do not exceed this value); the number of states in the HMM was fixed to 3, after some preliminary experimentation.

For making predictions we evaluate the different HMMs over the test sequence using the *Forward* algorithm, see [Rabiner, 1990] for details. We use the probabilities returned by each HMM as its confidence on the gesture class for a particular window.

3.3 Simultaneous segmentation and recognition

The multi-windows approach to gesture segmentation and recognition is as follows, see Figure 1. For processing a continuous video we trigger windows incrementally: at time t_0 a temporal window W_0 of length Δ is triggered and all of the (trained) HMMs are evaluated in this window. At time t_1 we trigger another window W_1 of length Δ and increase window W_0 by Δ frames, the HMMs are evaluated in these two windows too. This process is repeated until certain condition is met (see below) or until window W_1 surpass a maximum length, which corresponds to the maximum number of allowed simultaneous window, q.

In this way, at a time t_g we have $g-$ windows of varying lengths, and the outputs of the $K-$HMMs for each window (i.e., a total of $g \times K$ probabilities, where K is the number of gestures or activities that the system can recognize). The outputs of the HMMs are given in the form of probabilities. To obtain a prediction for each window i we simply keep the label/gesture corresponding to the model that obtains the highest probability in window i, that is, $argmax_k P(\mathcal{M}_k, W_i)$.

In order to detect the presence of a gesture in the continuous video we estimate at each time t_j the percentage of votes that each of the $K-$gestures obtains, by considering the predictions for the $j-$windows. If the number of votes exceeds a threshold, τ, we trigger a flag indicating that a gesture has been recognized. When the flag is on, we keep increasing and generating windows and storing predictions until there is a decrement in the percentages of votes for the dominant gesture. That is, end of the gesture happens in the frame where there is a decrement in the number of votes. Alternatively, we also experimented with varying the window in which we segment the gesture: we segmented the gesture 10 frames before and 10 frames after we detect the decrement in the percentage of votes, we report experimental results under the three settings in Section 4. At this instant the votes for each type of gesture are counted, and the gesture with the maximum number of votes is selected as the recognized gesture. Once a gesture is recognized, the system is reset; that is, all ongoing windows are discarded and a the process starts again with a single window.

One should note that the less windows we consider for taking a decision the higher the chances that we make a mistake. Therefore, we ban the proposed technique for making predictions before having analyzed at least $p-$windows. Under these settings, our proposal will try to segment and recognize gestures only when the number of windows/predictions is between (p, q).

Figure 2 illustrates the process for simultaneous segmentation and recognition for a particular test sequence containing one gesture. The first three plots show the probabilities returned by the HMMs for three gestures; we show the probabilities for windows starting at different frames of the continuous sequence. The fourth plot shows the percentage of votes for a particular gesture at different segments of the video. For this particular example, the proposed approach is able to segment correctly the gesture (the boundaries for the gesture present in the sequence are shown in gray). In the next section we report experimental results obtained with our method for simultaneous segmentation and recognition of gestures.

4 Experimental results

We performed experiments with the multi-windows approach by trying to recognize gestures to command a robot. Specifically, we consider three gestures: *move-right (MR), attention (ATTN), move-left (ML)*, these are illustrated in Figure 3. For evaluation we generated sequences of gestures of varying lengths and applied our method. The number of training and testing gestures are shown in Table 1. Training gestures were manually segmented. Test sequences are not segmented; they contain a single gesture, but the gesture is surrounded by large portions of continuous video without a gesture, see Figure 2.

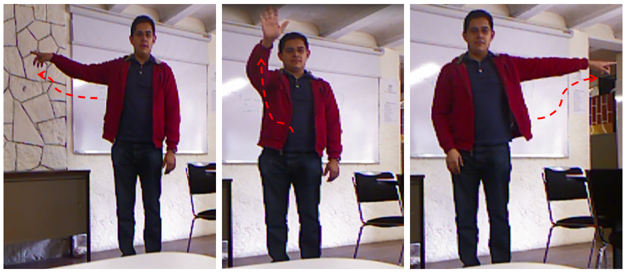

Figure 3: The three gestures considered for experimentation. From left to right: *move-right, attention, move-left*.

Three different subjects recorded the training videos. The test sequences were recorded by six subjects (three of which were different from those that recorded the training ones). The skeleton information was recorded with the NUI Capture software[3] at a rate of 30fps. The average duration of training gestures was of 35.33 frames, whereas the average duration of test sequences was of 94 frames (maximum and minimum durations were of 189 and 55 frames respectively).

All of the parameters of our model were fixed after preliminary experimentation. The better values we found for them are as follows: $\Delta = 10$; $p = 30$; $q = 60$; $\tau = 100$. After training the HMMs individually, we applied the multi-windows approach to each of the test sequences.

We evaluate the segmentation and recognition performance as follows. We say the proposed method correctly segments

[3] http://nuicapture.com/

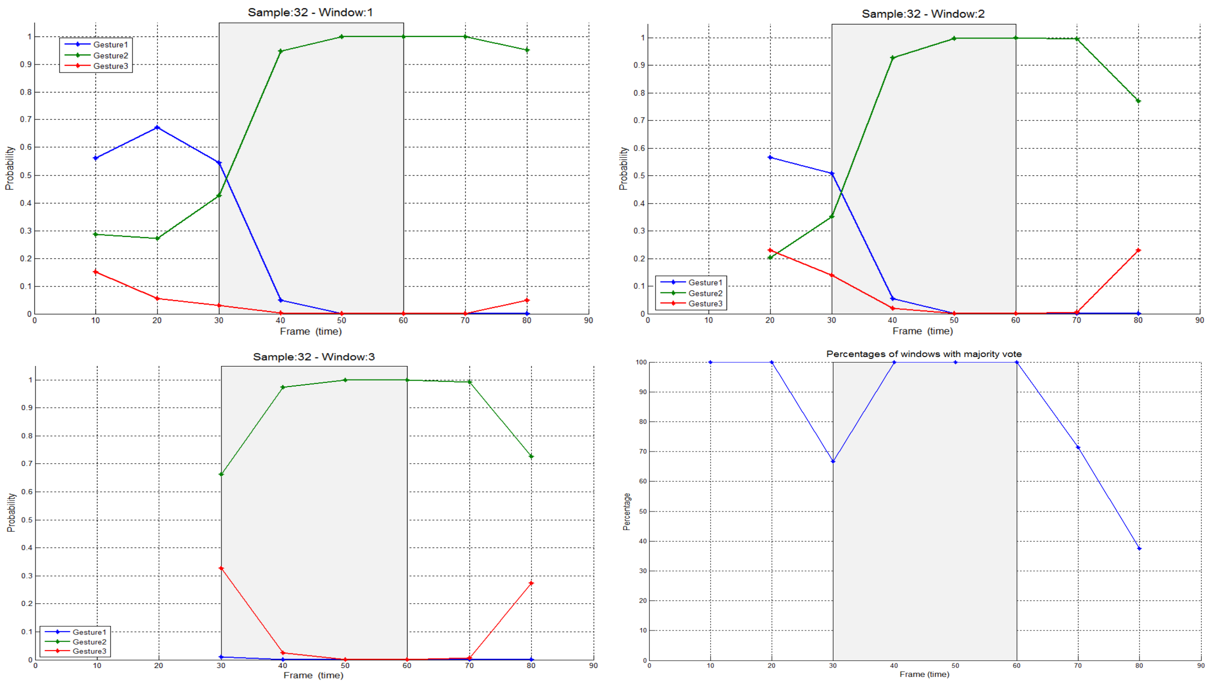

Figure 2: Multi-windows technique in action. The first three plots show probabilities obtained per each HMM for windows starting at different times. In the bottom-right plot we show the number of votes obtained by the dominant HMM, note that the number of votes start to diminish, this is taken as an indication of the end of the gesture (best viewed in color).

Table 1: Characteristics of the data set considered for experimentation. We show the number of training videos per gesture, and, in row two, the number of gestures present in the test sequences.

Feature	MR	ATTN	ML
Training vids.	30	30	30
Testing vids.	18	18	21

Table 2: Segmentation (**Seg.**) and recognition (**Rec.**) performance of the multi-windows technique. .

	Before		**In**		**After**	
δ	**Seg.**	**Rec.**	**Seg.**	**Rec.**	**Seg.**	**Rec.**
5	29.82%	82.35%	26.32%	60.00%	26.32%	80.00%
10	54.39%	67.74%	63.16%	66.67%	50.88%	68.97%
15	59.65%	64.71%	70.18%	67.50%	56.14%	68.75%
20	78.95%	62.22%	80.70%	63.04%	73.68%	66.67%

a video when the segmentation prediction is at a distance of $\delta-$frames (or less) from the final frame for the gesture; we report results for $\delta = 5, 10, 15, 20$. On the other hand, we say the proposed method correctly recognizes a gesture, when the gesture predicted by our method (previously segmented) was the correct one.

Table 2 shows the segmentation and recognition performance obtained by the multi-windows approach. We report results when segmenting the gesture before, in and after the decrement in percentage of votes is detected, see Section 3.

From Table 2 it can be observed that segmentation performance is low under a hard criteria (i.e., $\delta = 5$ frames of distance), the highest performance in this setting was of 29.82%. However, the recognition performance is quite high for the same configuration, achieving recognition rates of 82.35%. Thus, the method offers a good tradeoff[4] between segmentation and recognition performance.

In order to determine how good/bad our recognition results were, we performed an experiment in which we classified all of the gestures in test sequences after we manually segmented them (top-line). The average recognition performance for that experiment was of 85.96%. This performance represents the best recognition performance we could obtain with the features and trained models. By looking at our best recognition result (columns **Before**, row 1), we can see that the recognition performance of the multi-windows approach is very close to that we would obtain when classifying segmented gestures.

As expected, segmentation performance improves when we relax the distance to the boundaries of the gesture (i.e., for increasing δ). When the allowed distance is of $\delta = 20$

[4] Despite the fact that segmentation performance may seem low, one should note that for the considered application it is not too bad for an user to repeat a gesture 3 times in order that a robot correctly identifies the command we want to transmit. Instead, accurate recognition systems are required so that the robot clearly understand the ordered command, even when the user has to repeat the gesture a couple of times.

frames, we were able to segment up to 80% of the gestures. Recognition rates decreased accordingly. When we compare the segmentation performance obtained when segmenting the gesture before, in or after the decrement of votes, we found that the performance was very similar. Although, segmenting the gesture 10 frames before we detected the decrement seems to be a better option. This makes sense, as we would expect to see a decrement of votes when the gesture already has finished.

Regarding efficiency, in preliminarily experiments we have found the proposed method can run in near real-time. In a state-of-the art workstation, it can process data at a rate of 30fps, which is enough for many human-computer interaction tasks. Nevertheless, we still have to perform a comprehensive evaluation of our proposal in terms of efficiency and taking into account that in some scenarios a high-performance computers are not available.

From the experimental study presented in this section we can conclude that the proposed method is a promising solution to the problem of simultaneous gesture segmentation and recognition. The simplicity of implementation and the efficiency of our approach are beneficial for the development of ubiquious and human-centric systems.

5 Conclusions and future work directions

We proposed a new method for simultaneous segmentation and recognition of gestures in continuous video. The proposed approach combines the outputs of classification models evaluated in multiple temporal windows. These windows are dynamically and incrementally created as the video us scanned. We report preliminary results obtained with the proposed technique for segmenting and recognizing gestures to command a robot. Experimental results reveal that the recognition performance of our method is very close to that obtained when using manually segmented gestures. Segmentation performance of out proposal is still low, yet current performance is acceptable for the considered application. The following conclusions can be drawn so far:

- The proposed method is capable of segmenting gestures (with an error of 5 frames) at low-mild recognition rates. Nevertheless, these rates are accurate-enough for some applications. Recall we are analyzing a continuous sequence of video and that we do not require of a model for *no-gesture*, as required in related models.
- Recognition rates achieved by the method are acceptable for a number of applications and domains. In fact, recognition results were very close to what we would obtain when classifying manually-segmented gestures.
- The proposed method is very easy to implement and can work in near real-time, hence its applicability in ubiquitous data mining and human-centric applications are quite possible.

The proposed method can be improved in several ways, but it remains to be compared to alternative techniques. In this aspect we have already implemented the method from [Kim *et al.*, 2007], but results are too bad in comparison with our proposal. We are looking for alternative methods to compare our proposal.

Current and future work includes extending the number of gestures considered in this study and implementing the method in the robot of our laboratory[5]. Additionally, we are working in different ways to improve the segmentation performance of our method, including using different voting schemes to combine the outputs of the different windows.

References

[Aggarwal and Ryoo, 2011] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: a review. *ACM Computing Surveys*, 43:16(3), 2011.

[Aviles *et al.*, 2011] H.H. Aviles, L.E. Sucar, C.E. Mendoza, and L.A. Pineda. A comparison of dynamic naive bayesian classifiers and hidden. *Journal of Applied Research and Technology*, 9(1):81–102, 2011.

[Derpanis *et al.*, 2010] K. G. Derpanis, M. Sizintsev, K. Cannons, and R. P. Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *Proc. of CVPR*, pages 1990–1997. IEEE, 2010.

[Eunju, 2010] K. Eunju. Human activity recognition and pattern discovery. *Pervasive Computing*, 9(1):48–53, 2010.

[Huynh *et al.*, 2008] T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proc. of UbiComp'08*, pages 10–19. ACM Press, 2008.

[Kim *et al.*, 2007] D. Kim, J. Song, and D. Kim. Simultaneous gesture segmentation and recognition based on forward spotting accumulative hmms. *Pattern recognition*, 40(11):3012–3026, 2007.

[Malgireddy *et al.*, 2012] M.R. Malgireddy, I. Inwogu, and V. Govindaraju. A temporal bayesian model for classifying, detecting and localizing activities in video sequences. In *Proc. of CVPRW*, pages 43–48, 2012.

[Mitra, 2007] S. Mitra. Gesture recognition: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(3):311–324, 2007.

[Rabiner, 1990] L. E. Rabiner. *Readings in speech recognition*, chapter A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann, 1990.

[Webb and Ashley, 2012] J. Webb and J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apres, 2012.

[Yang *et al.*, 2007] H.D. Yang, A. Y. Park, and S. W. Lee. Gesture spotting and recognition for humanrobot interaction. *IEEE Transactions on robotics*, 23(2):256–270, 2007.

[Yuan *et al.*, 2009] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *Proc. of CVPR*. IEEE, 2009.

[5] http://ccc.inaoep.mx/ markovito/