2014 IEEE 26th International Conference on Tools with Artificial Intelligence

An HMM-based Gesture Recognition Method Trained on Few Samples

Vinicius Godoy, Alceu S. Britto Jr.,
Alessandro Koerich and Jacques Facon

Post-graduate Program in Informatics (PPGIa)
Pontifical Catholic University of Parana (PUCPR)
Curitiba (PR), Brazil
Email: {godoy, alceu, facon}@ppgia.pucpr.br

Luiz E. S. Oliveira

Post-graduate Program in Informatics (PPGInf)
Federal University of Parana (UFPR)
Curitiba (PR), Brazil
Email: lesoliveira@deinf.ufpr.br

***Abstract*—This paper addresses the problem of recognizing gestures which are captured using the Kinect sensor in a educational game devoted to the deaf community. Different strategies are evaluated to deal with the problem of having few samples for training. We have experimented a Leave One Out Training and Testing (LOOT) strategy and an HMM-based ensemble of classifiers. A dataset containing 181 videos of gestures related to nine signs commonly used in educational games is introduced, which is available for research purposes. The experimental results have shown that the proposed ensemble-based method is a promising strategy to deal with problems where few training samples are available.**

Keywords*—Gesture recognition; hidden Markov models; Kinect sensor.

I. Introduction

Creating an efficient method for gesture recognition of a sign language has been a big challenge for the Pattern Recognition community. The main reason is that there are hundreds of possible gestures and some of them are very similar to each other. In addition, gestures can be performed by different individuals at different speeds and captured at an uncontrolled environment which may present variations in the illumination, different objects within the scene, or even people in the background.

The nature of a sign gesture captured by a camera makes the Hidden Markov Model (HMM) [11] a very common approach to deal with this problem. Such statistical modelling has been used in CopyCat software, designed by Zafrulla et al. [12] to teach American Sign Language to children. In their research, a dataset with six signs created by two subjects, two prepositions and two objects was used. With the defined signs, 20 samples of 4 different sentences were captured using the Kinect sensor. They reported 98.8% of recognition rate, but considering just one subject.

HMM was also used in the work of Elmezain et al. [4] for recognition of Arabic digits (3 to 7) using a stereo BumbleBee camera. In their method, hands were segmented using a Gaussian based YCbCr color space skin detection algorithm. The feature extraction was based on the hands position and orientation. A high recognition rate of 98.4% was reported on 98 videos of isolated gestures (10 classes), but unfortunately the authors provided few information about the dataset used.

A recent contribution using HMM is reported in [3] for segmentation and recognition of gestures captured using Kinect sensor with the objective of commanding a robot. The reported recognition performance on 5 different gestures was 89.5%. Other recent results are reported in [5], where it is possible to observe HMM-based solutions on the highest ranking positions of a chalenge on gesture recognition.

Another machine learning algorithm commonly used to deal with such a challenging problem is the neural network. It can be found in the work of Carneiro et al. [2], in which Hu invariant moments are applied to extract six different hand-based features. Then, the result is applied to a Self Organizing Map (SOM) and the classification is carried out by two MLP (Multi-Layer Perceptron) neural networks. They classify 50 individual images of three different people, each one performing 26 gestures of the Arabic Alphabet in a total of 3,900 images. The authors reported recognition rates of 90% and 89.6%, when using Perceptron and MLP respectively.

Similarly, Paulraj et al. [10] used neural networks to classify 14 Malaysian sign language gestures captured with a 2D video camera with 320 $\times$ 200 resolution and 24 bit color depth. All frames were converted to grayscale and segmented considering three areas: head, left and right hand. The features were extracted using the Direct Cosine Transform (DCT) and classified by a neural network. The reported recognition rate using 98 videos for training and 140 for testing was 81.0% of accuracy.

Despite the use of different inputs (video or image), features, and classifiers, the aforementioned contributions share a common challenge: to guarantee high classification rates for such a complex task, usually, with a few samples for training. Gathering video or image to build a dataset is a time consuming and expensive task. With this in mind, the objective of this paper is three-fold. First, we address the problem of classifying a set of nine different gestures of a educational game for the deaf community, generating a list of three possible word candidates ranked according their likelihood. Second, we evaluate two strategies to deal with few samples for training: Leave One Out Training and Testing (LOOT) scheme [6] and an HMM-based ensemble

1082-3409/14 $31.00 © 2014 IEEE
DOI 10.1109/ICTAI.2014.101

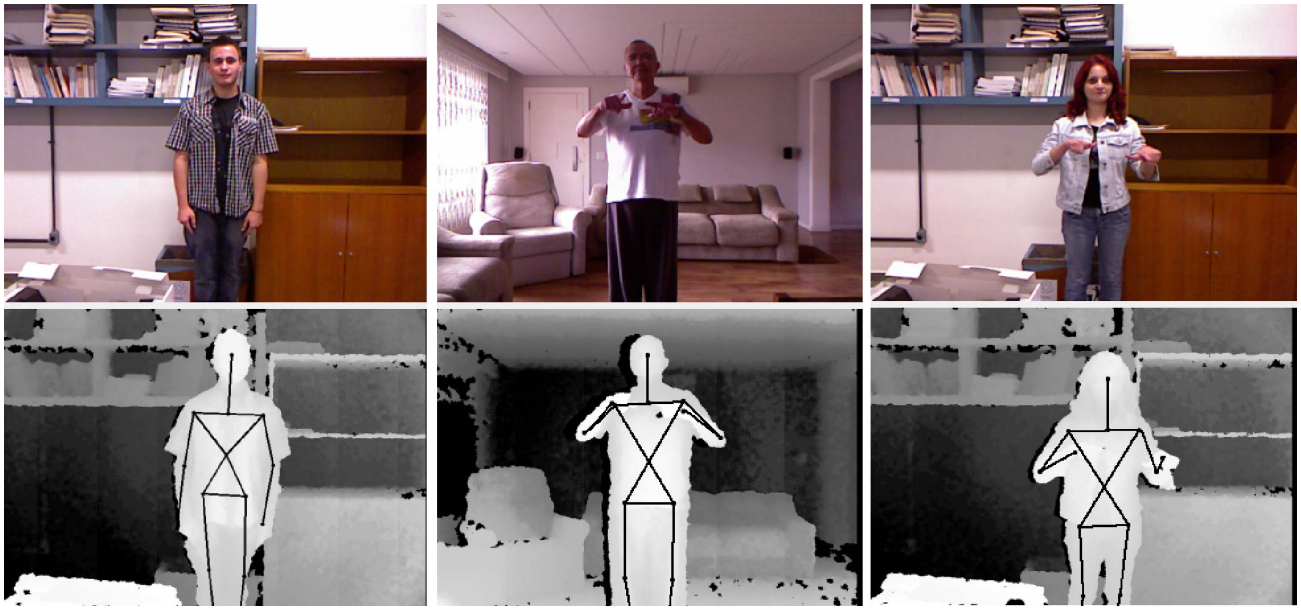


Fig. 1: Samples of the dataset: RGB and depth images with the corresponding skeletons for different subjects and scenarios

of classifiers. Finally, we present a video dataset containing 181 videos of gestures and the corresponding skeletons both provided by the Kinect sensor. The selected nine gestures (or words) are common in educational games devoted to the deaf community. This dataset is publicly available for research purposes at: http://bit.ly/librasdb.

The paper is organized as follows. After this brief introduction, Section II presents the created dataset. Section III presents the proposed method, describing the feature extraction and classification algorithms. Section IV describes the strategies used to deal with small training sets. Section V describes the experimental results, while in Section VI, we present our conclusions and insights for future work.

II. Dataset

We created a video dataset containing 9 different gestures of the Brazilian Sign Language (Libras), which are usually found in educational games of the deaf community. For this purpose, 23 subjects of different ages varying from 17 to 60 years were used. The subjects had different statures, sex and skin color. Each subject recorded up to three videos for each word. Some samples are shown in Figure 1.

Since Libras gestures are done using the primary hand, there are both right and left handed subjects in the dataset. It is important to notice that, although the main arm is flipped, the motion direction remains the same. All videos were captured with the Kinect sensor. For each gesture, the RGB video, the depth video and the skeleton provided by that sensor are available. The videos were recorded using 640×480 resolution and 30 frames per second. The RGB images were recorded without any compression, the 16 bit depth information was compressed using the 16ET lossless algorithm, while the skeleton was recorded using a binary format. A total of 181 videos and their corresponding artefacts were captured. Table I shows how they are distributed.

TABLE I: Number of videos and subjects per gesture

Word	# of videos	# of different subjects
Give (Entregar)	22	12
Take (Pegar)	20	11
Open (Abrir)	20	11
Look (Olhar)	18	12
Push (Empurrar)	20	14
Close (Fechar)	19	10
Talk (Falar)	23	14
Pull (Puxar)	19	8
Work (Trabalhar)	20	16

III. Proposed Method

Figure 2 shows an overview of the proposed method. As one may see the depth image and the skeleton provided by the Kinect sensor are the inputs of the proposed method. From the depth image, the left and right hands are segmented and the Scale-invariant Feature Transform (SIFT) is computed. The rationale behind that is to represent the different hand configurations, or hand-poses, inherent to the gestures of the Libras signs. In addition to the hand pose features, spacial-based features are extracted from the skeleton, which provides the trajectory of both left and right hand. A quantization process is used to create a codebook to map feature vectors to sequences of observations which are necessary when using discrete HMMs. The leave-one-out scheme and the classification based on ensemble of HMMs are used to deal with the problem of having few samples for training. The next subsections explain the main characteristics of the proposed method.

A. Hand detection

The hand detection on the depth image is done using the joint points available in the skeleton provided by the Kinect sensor. The skeleton consists of an array of joints with the

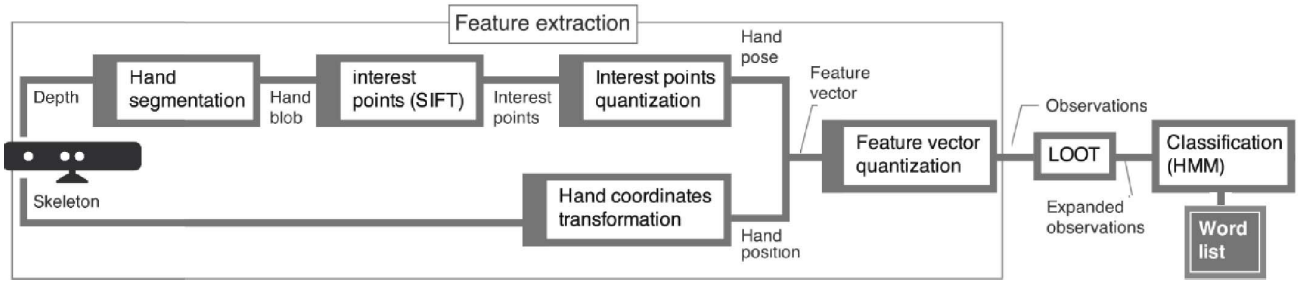


Fig. 2: Overview of the proposed method

position of twenty recognized human joints at each video frame. Figure 3 shows each joint available in the skeleton.

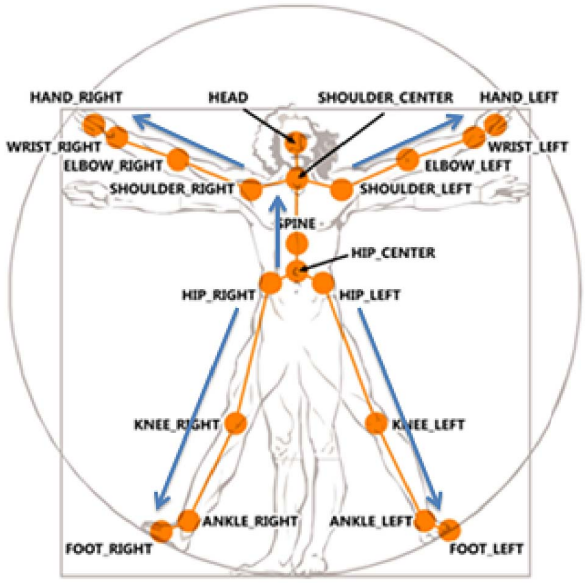


Fig. 3: The 20 recognized human joints provided by the kinect sensor, source MSDN Library [9]

The algorithm used for hand detection consists of two steps. First, the region of interest (ROI) of both hands are defined in the depth image based on their joint points, HAND_RIGHT and HAND_LEFT in Figure 3. Each ROI is defined as a square with 75×75 pixels centered at each hand skeleton joint point as shown in Figure 4.

The second step consists in detecting the hands inside the respective ROIs. To that end, an histogram of distances is constructed using the pixel values of the depth image. The first peak in this histogram represent the hand, which is usually in front of the scene. Figure 5 shows an example of the peak used to segment the hand from the background of the death image. A valley is found by isolating the first local maxima, and considering all pixels K centimeters below it (K was experimentally defined as 12). It is worth noting that the result is not a binary image, since the depth information is still available.

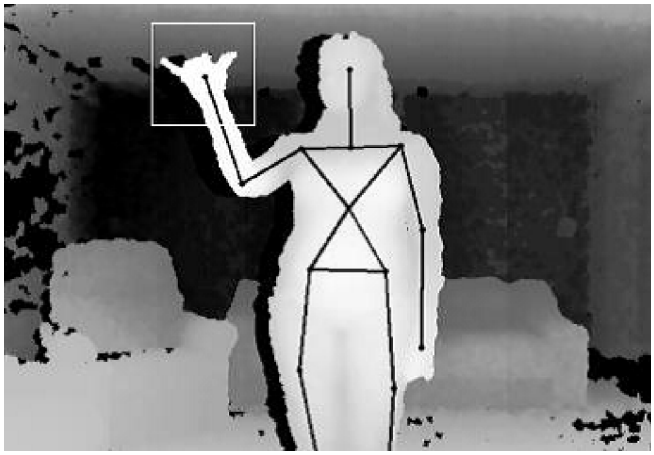


Fig. 4: Example of the ROI detected for the right hand using the skeleton information

B. Feature Extraction

There are two sort of features in the proposed gesture recognition method, pose and trajectory-based features.

- Hand pose-based features: after hand detection, the SIFT algorithm is used to detect its interest points, which are described by their location and orientation. SIFT was originally proposed by Lowe [7] to extract features from images to perform matching of different views of an object or scene. This algorithm uses scale-space Difference-of-Gaussian to detect interest points in images, and describe them by means of an interesting set of features. The extracted features are invariant to rotation, scaling and partially invariant to changes in illumination and affine transformation. In this work, SIFT provide us a set containing the coordinates and orientation of P interest points for each hand. By using the OpenCV implementation of this function on the hand depth blob, with its default parameters (3 nOctaveLayers, 0.04 contrast threshold, 10 border threshold and 1.6 sigma), we separate the P more significant points detected as interest points. The value of P was experimentally defined as 10. Figure 6 shows the hand segmented and the corresponding points of interest detected by the SIFT algorithm. The resulting 30-dimensional feature vectors (3 features

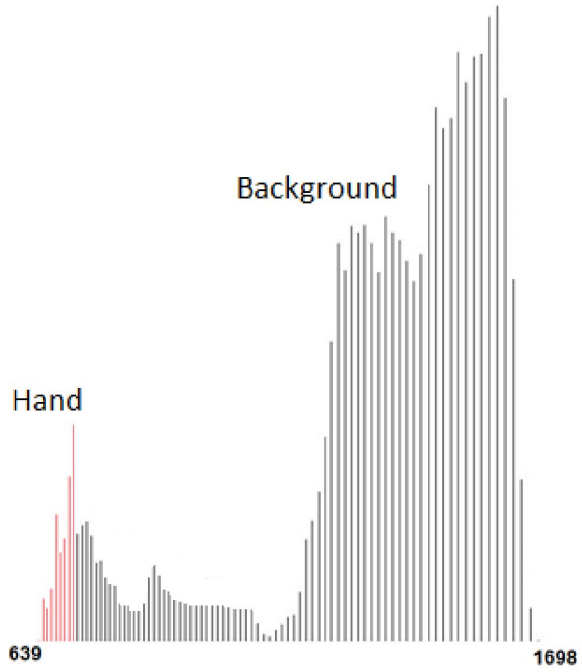
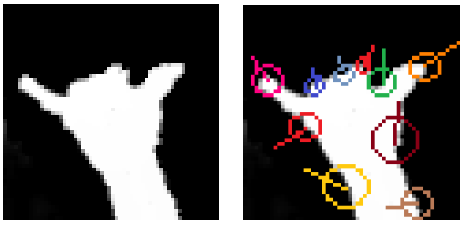


Fig. 5: ROI histogram used to hand segmentation from the image depth background



(a) (b)

Fig. 6: (a) Hand segmented, (b) Points of interest using SIFT)

x 10 points) are then quantized using the k-means algorithm, in order to discretize the hand pose in one of L codebook entries, representing the possible hand poses.

- Hand trajectory-based features: the left and right hand movements are automatically detected by the Kinect Sensor. The original coordinates are based on the camera position. The x and y axis are related to the camera center, and the z coordinate represents the distance between the object and the camera. This is not a suitable way to represent these points since people have different heights and the subject is not necessarily in the middle of the scene. To solve this problem, two transformations were applied. First, it is necessary to translate all skeleton points, in order to transform the hip joint (HIP_CENTER) as the coordinate system center. Figure 7 shows the adjustment of the system center based on the hip joint. Afterwards, let m and q be the hand and hip

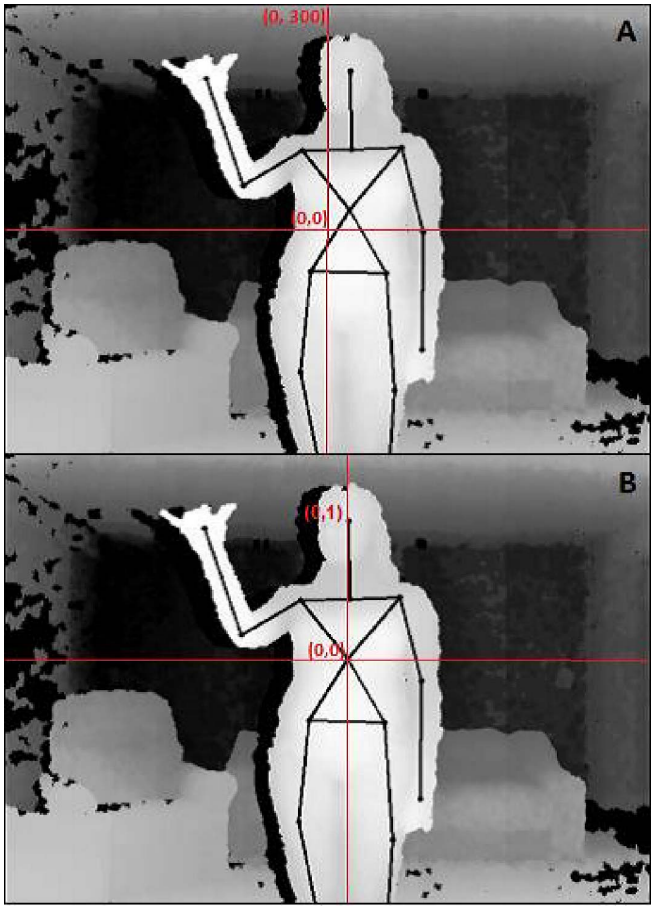


Fig. 7: Adjustment of the system center coordinates according to the hip joint available in the skeleton

position vectors. The transformed coordinate m_t can be calculated as depicted in equation 1. Second, a scale transformation is applied in order to normalize the people differences in height. This transformation is motivated by the observation that, although people have different size, the body proportions are basically the same[1]. Therefore, by using the hip to head height as a reference, it is possible to normalize all vectors, making them invariant to different statures. Let c be the head vector (HEAD in the Figure 3) in the camera space, we can calculate the scale transformation by dividing the transformed coordinate by the hip to head height using the equation 2.

$$m_t = m - q \tag{1}$$

$$m_f = \frac{m_t}{|(q - c)|} \tag{2}$$

Thus, the final feature vector contains four features (the three trajectory coordinates and the hand pose code) which are calculated for each video frame. As mentioned before, a codebook with M entries (experimentally defined as 33) is

constructed using the available training videos. This codebook is used to map the feature vector of a frame into a codebook entry, generating a sequence of discrete observations for each scene of a single gesture. It is important to notice that two different sequences are generated, one for each hand.

C. Classification

With the sequences of observations generated by the proposed feature extraction method, each gesture is modeled by at least two HMMs. We have used first-order discrete HMMs which can be described as depicted in the notation 3:

$$\lambda = \{N, M, T, A, B, \pi\} \tag{3}$$

where, N is number of distinct states in the model, M is the number of distinct observation symbols per state (it corresponds to the size of our second codebook), T is the length of a given observation sequence (we have two discrete observation by video frame, left and right hands), A is the state transition probability distribution, B is the observation symbol probability distribution at each state, and π is the initial state distribution. The last three parameters were learned by means of the Baum-Welch algorithm, while the Forward algorithm was used to estimate the probability of each model having produced a given sequence of observations. With respect to the HMM topology, ergodic and left-to-right models were used. A detailed description about HMM, the possible topologies and the corresponding algorithms are available in [11].

Two classification schemes were evaluated, a combination of one single HMM for each hand, and a combination of one ensemble of HMMs for each hand, as follows:

- Left and Right Hand Model-based (LRM): Here, two HMMs were trained for each of the nine gestures, one for each hand, as represented in equation 4.

$$LRM = \lambda^{left} \times \lambda^{right} \tag{4}$$

where,

$$\lambda^{left} = \left\{\lambda_1^{left}, ..., \lambda_9^{left}\right\} \tag{5}$$

and,

$$\lambda^{right} = \left\{\lambda_1^{right}, ..., \lambda_9^{right}\right\} \tag{6}$$

As one may see, the λ^{left} and λ^{right} models are combined to provide the final decision. The fusion scheme is the product of their probability. The topology and the parameters of the HMMs in the LRM approach, number of states and symbols per state (codebook size), were optimized by a genetic algorithm (GA). This optimization algorithm uses a binary genome, with 58 bits, distributed as follows: a) 4 bits for the codebook size, which allows a variation between 20 and 35; and, for each HMM: b) 1 bit for the topology (ergodic or left/right); and c) 5 bits for the number of states (5 to 36). The algorithm used asymptotic selection, as proposed in [8], a crossover probability of 0.75, mutation probability of 0.01 and elitist of a single individual. Table II presents the final topology and number of states of each gesture HMM.

- Left and Right Hand Ensemble-based (LRE): In this classification scheme, for each gesture, an ensemble of HMMs is created for each hand. Left and right ensembles of HMMs are combined to provide the final decision. Here, the fusion scheme ($\bigotimes$) is done in two stages. First, the corresponding left and right models are combined by the product of their probability and them the majority voting rule is used to provide the final ensemble decision. LRE notation is depicted in equation 7.

$$LRE = \lambda_{i,j}^{left} \bigotimes \lambda_{i,j}^{right} \tag{7}$$

where,

$$1 \leq i \leq 9 \tag{8}$$

and,

$$1 \leq j \leq Q \tag{9}$$

In the LRE approach, the expected diversity is ensured by generating HMMs based on different parameters. For this purpose, the topology, the number of states and symbols per state are varied to create for a given gesture Q HMMs for each hand, where Q were experimentally defined as 30. Also experimentally, we have observed the best results when the codebook size randomly varies from 20 to 35, and the number of states varies from 5 to 56.

TABLE II: LRM - final topology and number of states of each gesture HMM

Model	Topology	# of states
Give (Entregar)	left-to-right	23
Take (Pegar)	left-to-right	21
Open (Abrir)	left-to-right	22
Look (Olhar)	ergodic	23
Push (Empurrar)	ergodic	9
Close (Fechar)	left-to-right	23
Talk (Falar)	left-to-right	27
Pull (Puxar)	ergodic	19
Work (Trabalhar)	left-to-right	25

IV. The LOOT Strategy

As mentioned before, one of the main problems when dealing with gesture modeling may be the few number of training samples. The reason is that the generation of such video or image datasets is usually very expensive and time-consuming. To overcome this problem, we have evaluated here the leave-one-out training and testing (LOOT) strategy proposed in [6]. The basic idea behind such a technique is to generate secondary observation sequences from each original one, in order to reduce noise in the HMM training.

Let us to consider an observation sequence of length T as $O = \{o_1, o_2, \ldots, o_T\}$. We can generate a new sequence by removing one observation at each time from O, as demonstrated in Figure 8 for a sequence of length 5.

The new sequences can reduce or induce noise in training. If they reduce it, we will obtain a more reliable set of

Main sequence	$O = \{o_1, o_2, o_3, o_4, o_5\}$
Secondary sequences	$O = \{o_2, o_3, o_4, o_5\}$
	$O = \{o_1, o_3, o_4, o_5\}$
	$O = \{o_1, o_2, o_4, o_5\}$
	$O = \{o_1, o_2, o_3, o_5\}$
	$O = \{o_1, o_2, o_3, o_4\}$

Fig. 8: Generating new observation sequences - LOOT strategy

TABLE III: Accuracy (%) of the LRM classification approach with and without the LOOT strategy, and LRE approach with LOOT, considering different number of samples for training

# of training samples (for each word)	LRM without LOOT			LRM with LOOT			LRE with LOOT		
	top1	top2	top3	top1	top2	top3	top1	top2	top3
1	23.7	24.8	25.9	23.7	25.4	26.5	29.0	29.6	33.3
3	24.3	25.9	28.1	25.9	28.7	30.3	46.9	49.4	50.6
5	34.8	41.4	53.5	37.5	43.0	57.4	59.9	64.8	66.0
10	61.3	76.8	78.4	63.5	77.9	79.0	85.0	87.7	90.7
14	66.3	81.2	83.9	69.0	84.5	86.1	88.5	98.8	98.8
17	72.3	83.9	88.4	76.2	87.2	92.8	91.2	100.0	100.0

TABLE IV: Confusion matrix of the LRM approach (%)

Class	Give (Entregar)	Take (Pegar)	Open (Abrir)	Look (Olhar)	Push (Empurrar)	Close (Fechar)	Talk (Falar)	Pull (Puxar)	Work (Trabalhar)
Give (Entregar)	72.7	0	4.5	9.1	4.5	0	0	0	9.1
Take (Pegar)	0	65.0	0	0	0	0	0	35.5	0
Open (Abrir)	0	0	75.0	5.0	0	10.0	10.0	0	0
Look (Olhar)	0	0	11.1	83.3	5.6	0	0	0	0
Push (Empurrar)	0	0	10.0	0	85.0	0	0	0	5.0
Close (Fechar)	0	0	15.8	5.3	5.3	73.7	0	0	0
Talk (Falar)	0	0	8.7	4.3	4.3	0	82.6	0	0
Pull (Puxar)	0	31.6	0	0	0	5.3	0	63.2	0
Work (Trabalhar)	0	0	5.0	0	0	0	10.0	0	85.0

TABLE V: Confusion matrix of the LRE approach (%)

Class	Give (Entregar)	Take (Pegar)	Open (Abrir)	Look (Olhar)	Push (Empurrar)	Close (Fechar)	Talk (Falar)	Pull (Puxar)	Work (Trabalhar)
Give (Entregar)	100	0	0	0	0	0	0	0	0
Take (Pegar)	0	65.0	0	0	0	0	0	35.0	0
Open (Abrir)	0	0	100	0	0	0	0	0	0
Look (Olhar)	0	0	0	100	0	0	0	0	0
Push (Empurrar)	0	0	0	0	100	0	0	0	0
Close (Fechar)	0	0	0	0	0	100	0	0	0
Talk (Falar)	0	0	0	0	0	0	100	0	0
Pull (Puxar)	0	47.4	0	0	0	0	0	52.6	0
Work (Trabalhar)	0	0	0	0	0	0	0	0	100

observations for training. If noise is induced, the HMM will be trained to be more tolerant to noise on testing.

In order to use leave-one-out for testing, it is necessary to elaborate a strategy to combine the result of all generated sequences. Two approaches were proposed in [6], as follows:

- Hard approach: the first one consists in grouping the sequences by class result. Then, for each class, a new probability is generated by summing up all sequences associated to that class. This is similar to what is done in the SUM operator of a multiclassifier system. This approach is based on the assumption that all classifications will generate several errors, that can be minimized by the probability combination, or by voting. This is called the HARD approach, because all samples have great impact over the final result.
- Soft approach: the second consists in selecting the class with the best probability score directly. In conventional HMM, the objective is to find the class V_j with the highest probability among all classes $V_j, 1 \leq j \leq M$. Since the test generates T_i new classification sequences and since all observations have the same probability of being noise, we can assume that all sequences are equally important. Thus, by selecting the class with the highest probability score, this method allows that a sequence skips any individual observation to reach maximum possible proximity. The SOFT approach was used in our method.

V. Experimental Results

The focus of the undertaken experiments is to show the performance of the proposed classification methods by considering training sets with different number of samples. We have calculated the accuracies related to top1, top2 and top3 gesture

candidates. The top1 accuracy reflects the method performance when the correct gesture corresponds to its first candidate, top2 means that the correct gesture corresponds to its first or second gesture candidate, while in top3 the correct gesture is one of the three first gesture candidates. Such a scheme is interesting since in some applications, additional information obtained from the context can be used to make a final decision. A game is a special case in which the environment plays an important role.

The experiments were executed considering a rotation estimation scheme in which the objective is to leave p gesture samples out and use the rest of them for training. Since 18 is the lowest number of samples per gesture (see class Look in Table I), we have evaluated a set of p values in the range from 1 to 17. Table III summarizes the experimental results, while Tables IV and V show the confusion matrices related to the LRM and LRE methods, respectively.

As can be seen in Table III, the performance of both classification methods, LRM and LRE, shows high sensitivity to the number of samples for training. The LOOT strategy provides up to 4 percent points of improvement in the LRM classification accuracy. In addition, the ensemble-based method (LRE) surpassed the two model-based classification (LRM) in terms of recognition rate.

It is possible to observe that, LRE is also more robust for changes in the training size. For example, its top1 result for 10 videos for training (85.0%) is superior than the best result obtained by the LRM (76.2%).

The same main confusion is observed in the Tables IV and V, which is related to the gestures *Take* and *Pull*. The reason is that the hand trajectories of these gestures are the same, with a small diference in the hand pose. In this case, a possible solution is to use additional contextual information, which could be obtained by knowing the whole sentence or action of the subject.

VI. Conclusion and Future Work

We presented a method for classifying a set of gestures of a educational game. The method provides a list of candidate words ranked by their likelihoods. To create this method 181 videos were captured using the Kinect sensor and 23 different subjects.

The evaluated leave-one-out training and testing strategy provided an improvement of about 4 percent points in the classification performance. Beyond that, the proposed HMM ensemble-based method has shown to be an interesting approach reaching 91.2% of recognition rate for top1, while 100% was reached for top2. Further work can be done by adapting the algorithm to identify sentences instead of single words.

Acknowledgment

The authors would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) and the Research Foundation of the Parana state (Fundação Araucária).

References

[1] B. Bogin and M. I. Varela-Silva. Leg length, body proportion, and health: A review with a note on beauty. *International Journal of Environmental Research and Public Health*, 7(3):1047–1075, 2010.

[2] A.T. S. Carneiro, P. C. Cortez, and R.C.S. Costa. Reconhecimento de gestos da libras com classificadores neurais a partir dos momentos invariantes de hu. In *Interaction South America, Sao Paulo*, pages 190–195, 2009.

[3] H.V. Chavarria, H.J. Escalante, and L.E. Sucar. Simultaneous segmentation and recognition of hand gestures for human-robot interaction. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 25–29. IEEE, 2013.

[4] M. Elmezain, A. Al-hamadi, and B. Michaelis. A hidden markov model-based continuous gesture recognition. In *System for Hand Motion Trajectory, International Conference on Pattern Recognition (ICPR)*, pages 519–522, 2008.

[5] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H.J. Escalante. Chalearn gesture challenge: Design and first results. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–6. IEEE, 2012.

[6] A.H. Ko, P.R. Cavalin, R. Sabourin, and A.S. Britto Jr. Leave-one-out-training and leave-one-out-testing hidden markov models for a handwritten numeral recognizer: The implications of a single classifier and multiple classifications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2168–2178, 2009.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[8] V. G. Mendonca and C. T. Pozzer. A framework for genetic algorithms in games. In *VII Brazilian Symposium on Computer Games and Digital Entertainment*, pages 72–75, 2008.

[9] Microsoft Developer Network. Jointtype enumeration. *MSDN Library, http://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx*.

[10] M.P. Paulraj, S Yaacob, H. Desa, and W. Majid. Gesture recognition system for kod tangan bahasa melayu (ktbm) using neural network. In *Proc. of the 5th Int. Colloquium on Signal Processing and Its Applications*, pages 19–22, 2009.

[11] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[12] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proc. of the 13th Int. Conference on Multimodal Interfaces*, ICMI '11, pages 279–286, 2011.