Hand Gesture Recognition System using Hidden Markov Models

Ricardo Benítez Jimenez and Sergio Arredondo Serrano

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Sta. María Tonantzintla, CP 72840, Puebla, México.
{ricardo.benitez, sserrano}@inaoep.mx

Abstract. Ever since a couple decades ago, the processing power most computers were built with has progressed at such an impressive speed. One of the so many effects this processing increment has had is to shift the way most people perceive technology, in comparison to two decades ago. Nowadays, there is work being done on the development of computer-interfaces that facilitate the way we interact with computers. One interesting area is the development of interfaces for people with special needs, such as deaf people. Thus, in this document we present a gesture recognition system focused on the recognition of some Mexican Language signs, developed with computer vision techniques and probabilistic graphical models.

Keywords: Gestures Recognition, Hidden Markov Models, Hand Sign Language

1 Introduction

Nowadays, most of the people hold a really close relationship with technology, almost independently of what their daily activities are or how old they are, nevertheless, this has not always been the case. For instance, several decades ago they only people who had access to a computer were researchers in a few universities and laboratories, given that these kind of machines were very expensive and hard maintain. However, thanks to the digital revolution, which started around the 60's, since then the relation most people hold with technology could not be closer. Nowadays, is so common to find terms like "network", "pixel" or "RAM memory" in conversations of people who have never had a formal education on computer sciences, engineering or any similar discipline, and this is thanks to how normalized computers have become in the past few years. Even back when the first personal computers came out to the market at an accessible price for the regular folk, it was not really clear the use it could have in the common household, other than for generating spreadsheets. Since then, a lot of work has been done to facilitate the way people interact with computers. The faster these devices got every year, thanks to the advances made in the compression of electronics, it was possible to design more sophisticated interfaces which contributed to the popularization of computers, until it reached the current state we all know. In this way, it is safe to say the computer interfaces have a huge impact in the way general public perceives technology, that is, as something approachable and that we rely upon, even without understanding precisely the mechanism behind the screens that make these devices work.

Furthermore, a good indicator of the fast growing phase the computer-interface area has is how wide is the range of technologies that are being integrated to interfaces. For instance, haptic technologies which enrich the sense of touching by giving feedback to the user in the form of forces and vibrations. Additionally, another interesting technology that has been integrated to the interface of many devices, most of them developed for entertainment purposes, is computer vision techniques. Examples of this technology range from a simple face detector in a security system, all the way to a skeleton tracking system used in video consoles. The last of the aforementioned instances, is an example of a gesture detection system, which are particularly interesting interfaces because they have the capability of recognizing many more commands than a face detection interface, for instance, given that they combine information along time. Thus, in this document we present a system for the recognition of hand gestures, with an RGB camera as only sensor and using the Hidden Markov Models to perform the task of recognition of gestures.

The rest of the document is organized as follows. In section 2 we summarize the main advances done in the use of probabilistic models, such as the Hidden Markov Models, on applications related to gesture recognition. Section 3 describes in detail the techniques we employed for the development of our system. In section 4 we illustrate the experiments performed in order to evaluate the performance of our system. Finally, in section 5 the conclusions and future work are presented.

2 Related work

The task of recognizing gestures via an imagery sensor, such as a video camera, is multi-disciplinary problem that demands the integration of technologies such as computer vision, which perform the initial feature extraction procedure that results in the generation of discrete sub-gestures or observations that the recognition mechanism is able to understand and eventually use as inputs in order to predict a gesture. In terms of computer vision, the method used for gesture recognition applications mostly depend on the specifics of the application and the other techniques that are part of the system. For instance, in [1] a method based solely on the extraction of a histogram of local orientation from a gray scaled incoming stream is proposed for the recognition of simple hand movements, which uses an instance based learning approach for the recognition phase. In [2], in a similar fashion as the previous mentioned work, they use a template matching approach for the gesture recognition task, however, for the feature extraction phase a kinect sensor [3], for which they fusion the depth and color imagery, obtaining as a result a robust detection that is reflected in their 90.6% of accuracy. In [4] they use an appearance model approach for the features extraction task and for the gesture recognition a Hidden Conditional Random Field is used because this model does not make independence assumptions between the observations. They evaluate their proposal against a Hidden Markov Model approach, which they surpassed with a difference of 9.59% in accuracy. Furthermore, in [5] we found another example of the use of multiple data streams for the feature extraction task, however, in this case they used the earth mover's distance [6] as a similarity criteria along with a thresholding approach for the gesture recognition.

Moreover, for the problem of gesture recognition several approaches have been applied successfully, ranging from statistical modeling all the way to deterministic methods. Some examples of a deterministic technique used in recognition are found in [7,8], where finite state machines have successfully been employed for the recognition of several human gestures. Nevertheless, given that gestures are typically performed people as a form to communicate a message in a quick and easy way, these are performed with many involuntary variants, therefore, the most used methods for the recognition of human gestures are those that are good handling uncertainty. For instance, Kalman filtering [9], particle filtering [10] and condensation algorithms [11]. However, the Hidden Markov

Models (HMM) [12] , which are a probabilistic graphical model, are a kind of models that suit very well the problem of recognizing gestures, given that they were designed for the recognition of sequences of discrete objects, known as *observations* in the context of HMM. There are plenty of examples of HMM applied in this problem [13,14,15,16]. For this reason, we decided an HMM was the best alternative for our hand gesture recognition system, which is described in detail in th following section.

3 Methodology and development

3.1 General Architecture

Given that gesture recognition is a multidisciplinary field of study [17], it is necessary to integrate techniques and methodologies from several areas in order for an application to show a good performance. Of course, the set of disciplines involved in development process will vary among different cases of study. In this way, the methods incorporated into the system we present in this document belong namely to two areas of research: *computer vision* and *probabilistic graphical models*. Thus, our system can be seen as a process constituted by a sequenced arrangement of two main modules: The first one is an *Image Processing Module* (IPM), which performs all the operations relevant to sensing, sampling, pre-processing and feature extraction from the input data (images), and yields discrete values that the next module is able to identify. Subsequently, the *Gesture Recognition Module* (GRM) receives as input the discrete objects generated by the previous module, from which it builds a sequence of objects and will eventually classify into one of the known gestures. In figure 1 is shown the general scheme that defines our system's workflow process.

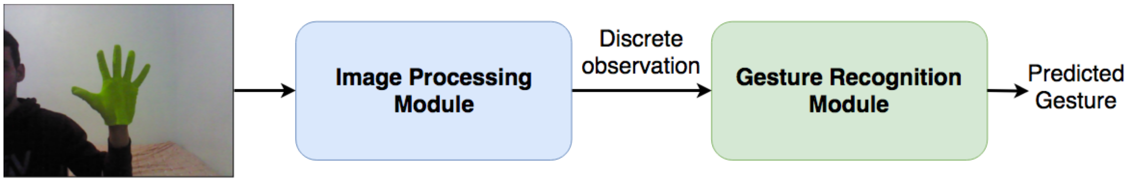

Fig. 1. Workflow process for the gesture recognition system.

3.2 Image Processing Module

As previously mentioned, the IPM is in charge of extracting from the raw images a discrete representation that the GRM is able to identify. In order to do so, the IPM performs a series of operations over the incoming images. Some of these operations apply transformations with the purpose of highlighting features that are relevant for the gesture recognition process, *e.g.* color segmentation or border detection. In addition to these, the feature extraction operations are used to yield from an image a discrete representation of some object that is of our interest, for instance, detecting the presence of a hand (which is a boolean variable) or counting how many fingers the user has extended (integer variable). Thus, below we describe the sequence of operations that constitute the IPM of our gesture recognition system.

1. **Sampling**: A sampling frequency of 4 Hz is used, in order to maintain a balance between temporal resolution and computing load that enables the system to operate in real time.

2. **Segmentation**: Given that the objective of our system is to recognize a limited set of hand language gestures, all of which are constituted by the action of a single hand, we decided to enable the user with a green glove that simplifies the task of identifying the pixels that are part of the hand by performing a color segmentation, which turned to be quite robust.

3. **Morphological transformation**: After the segmentation, a sequence of morphological operators (*opening*, *closing*) are applied in order to suppress small blobs originated form noise, yielding a clean binary image that only contains a blob corresponding to the hand.

4. **Hand characterization**: In this step, the blob that corresponds to the hand is characterized by detecting the center coordinate of the palm, counting how many fingers are extended and measure the angle between the horizontal axis and each vector that starts in center of the palm and ends on the tip of an extended finger. By the end of this step, the systems has determined if one of the 4 recognizable hand signs (see figure 2) is being displayed.

5. **Extract observation**: As a final step in the IPM, the discrete *observations* are yielded. The way this is done is by computing a shift vector from the previous and current coordinate of the center position of the palm, as seen in figure 3. Then, the angle with respect to the horizontal axis of this shift vector is compared to each of the 8 angles shown in the rightmost of figure 3. Depending on which of the 8 angles the shift vector is closest to and on which of the 4 static signs (see figure 2) is being displayed, will determine which *observation* is emitted by the IPM. Therefore, given that the *observation* depends on two discrete variables: the static hand sign and direction of the hand's movement, with 4 and 8 possible values respectively, this derives in 32 possible *observations*, plus an extra one that is generated when there is no movement (indistinctly of the hand sign), adds to a total of 33 possible *observations.*

As a side note, the whole IPM was implemented using the C++ programming language, along with the 2.4.8 version of the OpenCV library [18].

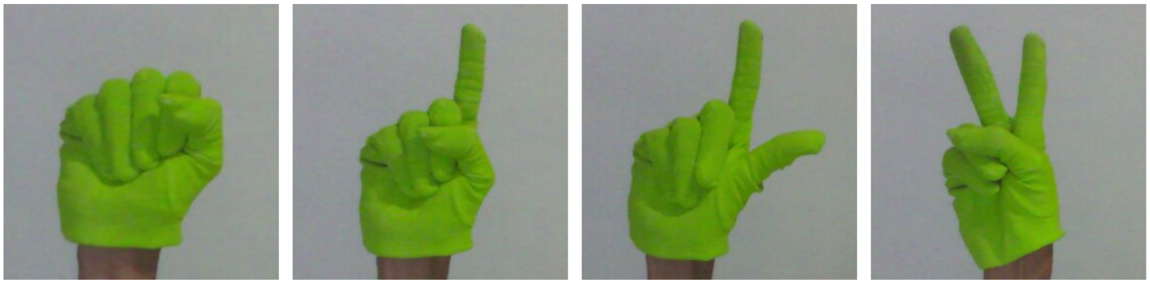

Fig. 2. Set of recognizable static hand signs (from left to right): *fist, index, "L" sign* and *peace.*

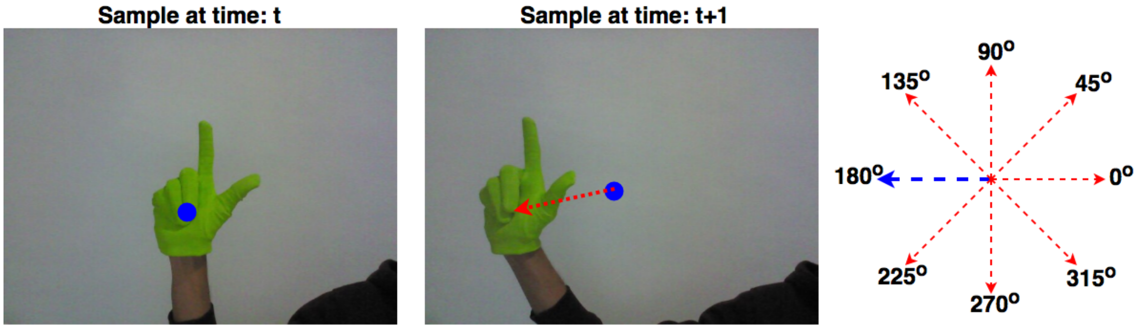

Fig. 3. A shift vector is computed between the position of a hand at two successive sampling instants and discretized into one of the 8 possible directions. In this example, a shift to the left and slightly down is discretized as a 180° vector.

3.3 Gesture Recognition Module

For the gesture recognition phase, we decided to integrate a Hidden Markov Model (HMM). This is a probabilistic graphical model [19] that suits very well tasks related to the classification of sequences. In our case, the sequences to be classified are constituted by the *observations* yielded in the final step of the IPM, that correspond to moving a hand while displaying one of the static hand signs shown in figure 2. Thus, before our system is able to recognize any sequence, it is necessary to define a finite set of known gestures we want to identify. Therefore, we defined a set of 5 gestures that correspond to the words *Viernes, Sábado, Lunes, Domingo* and the letter *Z*, in Mexican Sign Language[1], which are shown in figure 4. Furthermore, the classification of a sequence of *observations* is performed by applying the *forward algorithm* [20] on several trained HMM, one for each gesture in figure 4, and the gesture whose HMM has the highest posterior probability will be predicted class.

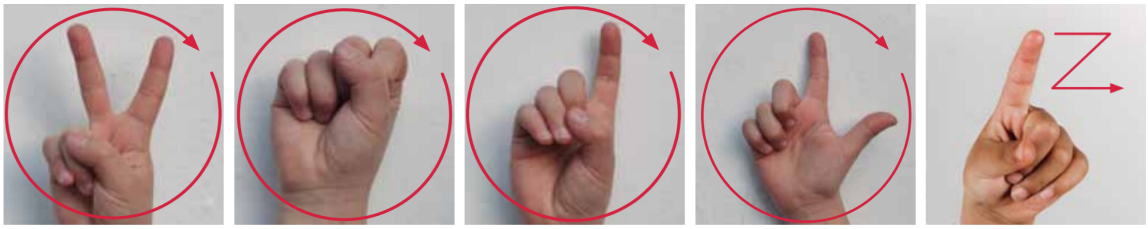

Fig. 4. Mexican Sign Language gestures for *Viernes, Sábado, Domingo, Lunes* and the letter *Z*.

[1] http://www.conapred.org.mx/documentos_cedoc/DiccioSenas_ManosVoz_ACCSS.pdf

4 Experiments and results

For each gesture, a Hidden Markov Model (HMM) was trained through the Baum-Welch algorithm which learns the transition and emission matrices from the training data, using a MATLAB implementation. Furthermore, although in the introductory section of this document it is stated that this gesture recognition system was developed towards the identification of Mexican Language signs, we decided to add 2 extra gestures: *Puerta* and *Ventana*, whose *observations* are generated by moving the *index* static hand sign, just like *Domingo* and *letter z*. These gestures were included in the evaluation phase with the purpose of analyzing how well HMMs perform when there are several sequences made of the same set of *observations*, which is the case with *Domingo, letter z, Puerta* and *Ventana*.

Thus, the training data consists of 40 examples for each gesture, 280 examples in total. To determine the number of states for each gesture, previous tests were performed, where the number of states was varied from 1 to 8. For each test, a 5-fold-cross-validation was used and the dataset was divided as follows: 32 examples were used to learn the transition and emission matrices, 8 examples were used to evaluate the HMM, the HMM's results were compared in terms of accuracy and the average probability when a correct predictions were made. In general, the configuration of four states obtained the best results or, at least, always ranked between the best two. Thus, in order to have a comparative framework, a number of four states was defined for the HMM of all the gestures.

In addition to the training examples, a test dataset was built with 20 examples for each gesture in total 140 examples. The main goal of this first test is to evaluate how well the HMM of each gesture recognizes its gesture and how well discriminate others gestures. In table 1 we can observe that all the HMMs make a good prediction of their gesture, but sometimes *letter z, Ventana* and *Domingo* are confused, given that these gestures have some *observations* in common, however, they discriminate well others gestures.

Table 1. Evaluation of HMMs for each gesture

Gesture.	TP	TN	FP	FN	Avg. log(prob) of TP	Accuracy	Precision	Recall	F1-Score
z	20	78	42	0	-13.38	0.7	0.322	1	0.487
Viernes	20	120	0	0	-41.76	1	1	1	1
Sábado	20	120	0	0	-9.2511	1	1	1	1
Lunes	20	120	0	0	-7.677	1	1	1	1
Domingo	20	96	24	0	-9.928	0.828	0.454	1	0.625
Puerta	13	120	0	7	-5.55	0.95	1	0.65	0.787
Ventana	18	107	13	2	-10.5	0.892	0.58	0.9	0.705
Avg.	18.71	108.71	11.28	1.28	-13.97	0.91	0.765	0.935	0.8

As we can see the HMM corresponding to *Viernes*, detects in average with the lowest probability the gestures compared with other HMMs, for instance, the HMM of *Puerta* which detects the gestures with the highest probability.

A second experiment was performed with the same test dataset, in which we defined a threshold with the intention of improving the performance of each HMM, by reducing the amount of *false positives*. The thresholds were defined by considering the minimum of the probability when approximately 100% of the examples were correctly classified and when the *false positives* started to appear. The gestures for which a threshold is required are *Domingo*, *z*, *Puerta* and *Ventana*. The thresholds were tested by adding steps of -10 all the way to -100. Afterwards, from all the values tested, the one with the largest amount of *true positives* and *true negatives* was refined with increments of -1. Finally, the threshold found for *z* and *Ventana* are -19 and -16, respectively. In the case of *Domingo* and *Puerta* a threshold was not found with this technique, however, in table 2 it is possible to see that by using the found thresholds, the gestures *z, Domingo* and *Ventana* improve in average, Accuracy, Precision and F1-Score.

Table 2. Evaluation of HMMs in test dataset with threshold

Gesture.	TP	TN	FP	FN	Threshold	Avg. log(prob) of TP	Accuracy	Precision	Recall	F1-Score
z	19	120	0	1	-19	-8.32	0.992	1	0.95	0.974
Viernes	20	120	0	0	0	-41.76	1	1	1	1
Sábado	20	120	0	0	0	-9.2511	1	1	1	1
Lunes	20	120	0	0	0	-7.677	1	1	1	1
Domingo	20	120	0	0	-18	-9.928	1	1	1	1
Puerta	13	120	0	7	0	-5.55	0.95	1	0.65	0.787
Ventana	18	120	0	2	-16	-10.5	0.985	1	0.9	0.947
Avg.	18.57	120	0	1.42	-13.97	-7.57	0.989	1	0.928	0.958

5 Conclusions and future work

From the differences in performance observed in tables 1 and 2 it is easy to observe that by defining a threshold for the posterior probability of some gestures, in general, the overall performance is improved. However, we would not guarantee that this threshold approach would improve the performance on a larger set of gestures. Moreover, the number of states in the HMM influences on the probability of a sequence is detected, however, one important thing is to determine the set of observations, in this work thanks to the representation of the observations was possible to get good results even with the same number of states on each HMM.

In this document, we presented a system for the recognition of a set of gestures that are part of the Mexican Language of signs. Although the system adequate for operation, there are some aspects that could be improved, for instance, there is a lot of work to do on the robustness of the feature extraction procedure as well on generating larger datasets that capture the variety of ways a single gesture can be performed.

References

1. William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, pages 296–301, 1995.

2. Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 759–760. ACM, 2011.
3. Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.
4. Sy Bor Wang, Ariadna Quattoni, L-P Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1521–1527. IEEE, 2006.
5. Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1093–1096. ACM, 2011.
6. Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.
7. Aaron F Bobick and Andrew D Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on pattern analysis and machine intelligence*, 19(12):1325–1337, 1997.
8. Pengyu Hong, Matthew Turk, and Thomas S Huang. Gesture modeling and recognition using finite state machines. In *Automatic face and gesture recognition, 2000. proceedings. fourth ieee international conference on*, pages 410–415. IEEE, 2000.
9. Greg Welch and Gary Bishop. An introduction to the kalman filter. department of computer science, university of north carolina. *ed: Chapel Hill, NC, unpublished manuscript*, 2006.
10. Cody Kwok, Dieter Fox, and Marina Meila. Real-time particle filters. In *Advances in neural information processing systems*, pages 1081–1088, 2003.
11. Michael Isard. Condensation. conditional density propagation for visual tr, king. *hit.. J. Coin put,. Vis*, 29(1):5–28, 1998.
12. Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
13. K Martin Sagayam and D Jude Hemanth. Abc algorithm based optimization of 1-d hidden markov model for hand gesture recognition applications. *Computers in Industry*, 99:313–323, 2018.
14. Jinxing Yang, Jianhong Pan, and Jun Li. semg-based continuous hand gesture recognition using gmm-hmm and threshold model. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 1509–1514. IEEE, 2017.
15. Yukun Dai, Zhiheng Zhou, Xi Chen, and Yi Yang. A novel method for simultaneous gesture segmentation and recognition based on hmm. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2017 International Symposium on*, pages 684–688. IEEE, 2017.
16. Nicolas Granger and Mounîm A el Yacoubi. Comparing hybrid nn-hmm and rnn for temporal modeling in gesture recognition. In *International Conference on Neural Information Processing*, pages 147–156. Springer, 2017.
17. Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.
18. Gary Bradski and Adrian Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 3, 2000.
19. Luis Enrique Sucar. Probabilistic graphical models. *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, 10:978–1, 2015.
20. Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE signal processing letters*, 10(1):11–14, 2003.

6 Appendix

For more detail on the code used to implement this project go to `https://bitbucket.org/ssergio/pgm/src/master/`.