Computer Science • 23(2) 2022 https://doi.org/10.7494/csci.2022.23.2.3745

Domínguez Ramón Emmanuel
Díaz Hernández Raquel
Altamirano Robles Leopoldo

HUMAN GESTURE RECOGNITION USING HIDDEN MARKOV MODELS AND SENSOR FUSION

Abstract *Considering the continued drive of human needs along with the constant improvement of technology, it is convenient to develop techniques that can enhance communication between computers and humans in the most intuitive ways possible. The possibility of automatically recognizing human gestures using artificial vision (among other kinds of sensors) allows us to explore a whole range of applications to control and interact with environments. Nowadays, most approaches for gesture recognition using sensors agree in the use of vision, myography, and movement devices that are applied to robotic, medical, and industrial applications. In the context of this work, we study the principles of using both vision and body contact sensing applied to the automatic classification of a human gesture set. For this, two different approaches have been evaluated: feed-forward neural networks, and hidden Markov models. These models have been studied and implemented for recognizing up to eight different human hand gestures that are commonly applied in collaborative robotics tasks.*

Keywords gesture recognition, collaborative robotics, vision-based sensors, myography sensors, sensor fusion

Citation Computer Science 23(2) 2022: 225–243

1. Introduction, motivation

Over the past years, human gesture-recognition research has been improved mainly due to the emergence of new devices that are aimed at creating new intuitive methods of interaction between humans and computers through human actions that are commonly performed by the hands and limbs and remapping them to understandable commands for high-level applications (such as robotics systems) in multiple areas.

In this sense, the automatic gesture-recognition process has been defined in [14] as a set of techniques that are aimed at representing and classifying different signals of communication between human users and work stations with defined objectives. In these processes, a user executes nonverbal commands by generally using his/her arms and hands to send specific messages according to the environment and intention for a high-level application. These result in the generation of human-machine interfaces.

Since the late 80s, users have mostly interacted by using only a keyboard and mouse through graphic interfaces. However, multiple devices have emerged over the last decade that have improved the methods for creating new and more intuitive interfaces (which are known as natural user interface systems). This new research topic aims at designing and implementing software systems efficiently and, above all, intuitively; this allows us to take full advantage of the potential benefits that are offered by computers and machines.

Nowadays, the most commonly used sensors for this purpose are based on computer vision techniques, electromyography (EMG), electroencephalography (EEG), and electrocardiogram (ECG) sensing, and movement and inertial sensors according to [9, 15]. In combination with computational signal-processing algorithms and new machine-learning approaches for pattern recognition, these techniques have many applications (such as home and office controlling [10] and industry intelligent systems [15, 25]). For this last field, the majority of the approaches are focused on human-robot collaborative activities and robotic arm manipulation.

In this context, we are particularly interested in the study and development of techniques for gesture recognition and their applicability for intuitive industrial and medical telemetry systems using multiple sensors. Several authors have addressed this issue [4, 7, 25]; they agree in regard to the use of wearable sensors and computational algorithms that involve temporal information along with collected data from devices. Wang et al. [25] reported high accuracy recognition rates for a set of up to six hand gestures; however, the use of muscle sensors without any other support restrains the quality of the recognition in some cases, misleads in similar gesture cases, and limits the numbers of total groups during the classification stages.

The objectives of this paper are mainly two. First, we propose a set of human hand and arm gestures that are capable of being incorporated into industrial robotics, medical telemetry, and collaborative robotics (among other applications). Second, developing and implementing sensor-fusion techniques and machine-learning algorithms in order to recognize up to eight different human gestures with high precision rates;

for this, we consider the use of two sensors: *leap motion controller* and *Myo armband* devices.

The following sections of this paper are structured as follows: Section 2 describes the most relevant approaches that have been proposed in the last years about automatic human gesture recognition. Subsection 2.4 briefly details some of the most recent applications of human gesture recognition. In Section 3, we describe the proposed approach for recognizing a whole set of gestures for robotic and collaboration applications. Our experiments and the results of their effectiveness over the proposed data sets are reported in Section 4. Finally, we discuss the potential applications and the contributions of this paper in Section 5.

2. Related work

2.1. Human gesture recognition

The studies that were made by Athavale and Deshmuk in [3] affirmed that human communication is 35% verbal-based and 65% non-verbal (and based on gestures). In this work, we focus on developing recognition techniques for the latter group; these can be divided as communicative (whose intention is to express emotions) and manipulative (which are aimed at expressing instructions and commands). In [16], Singh described that direct manipulation refers to methods that are aimed at controlling a specific variable through a user feedback interface in order to adjust the parameters of the first one (for example, a visual pointer that is related to position and movement in real time and remapped from the position and speed of a mouse or other input device).

Nowadays, multiple works have been proposed with the objective of automatically identifying manipulative gestures using sensors and machine-learning techniques. According to [27], the most recent approaches that are aimed at recognizing human gestures that are performed by the hands and limbs are supported by image-processing devices as well as wearable sensors that are capable of collecting information from the human body. The majority of these methods are composed of three main stages: modeling, analysis, and recognition.

The modeling stage consists of proposing the gestures to be studied according to the context in which they will be used. In this context, several researchers have studied and applied a variety of gestures for specific tasks. In [22], Rautiainen proposed a set of instructions that were based on human gestures in order to control multiple devices in the home and office. Another example of gesture sets was proposed in [25], where the authors studied the capabilities of EMG sensors that are applied to industrial applications; their proposed collection was the six hand-gesture set that is depicted in Figure 1. In [11], Nurettin provided a gesture vocabulary that was composed of 11 patterns to be applied to handle communication between humans and computers in intuitive ways by using movement sensors.

Other applications that are aimed at controlling aerial vehicles with simple hand commands were proposed in [17].

Figure 1. Human hand-over intentions for human-robot collaboration by [25]: (a) to need; (b) to give; (c) to stop; (d) to continue; (e) speed up; (f) slow down

The analysis process focuses on the feature extraction for each of the considered gestures that permits identifying them as unique in the following stages. For this purpose, movement and spatial features were considered in [11,24] in order to characterize gestures from a human's arms. With the same objective, Jorgensen proposed a computation stage of three-dimensional descriptors and *fast point feature histograms* (FPFH) in [8] according to movements that are performed by a user to describe eight different intentions by using depth cameras. In [24], a method for combining information from a *Kinect* sensor and EMG devices applying *weighted D-S* evidence theory processes from limbs and hands was proposed to identify two human gesture sets. However, the high computational resources of these techniques restrained their application in multiple real industry tasks.

Finally, the recognition stage is aimed at implementing computational algorithms and machine-learning techniques that allow for the automatic identification of human gestures by using the previous computed features. In this sense, several authors have addressed this problem by applying a variety of computational resources. In [21], Rautaray proposed the use of *hidden Markov models* (HMM) to recognize a set of human gestures and developed methods for processing and classifying commands from the hands that were targeted for home automation applications. Other approaches have also used HMM for gesture recognition in [19,23]; more recently, Wang addressed the recognition of six different hand gestures through myography wearable sensors in [25] and obtained good accuracy recognition rates.

The complete gesture set that was proposed is depicted in Figure 1. In [6], the authors used physical contact sensors to recognize hand gestures. In [12], the authors used EMG and IMU sensors.

In this work, we retake the most relevant gestures that have been proposed in order to develop techniques that are aimed at recognizing a greater number of human commands by using a combination of both myography wearable and vision-based sensors.

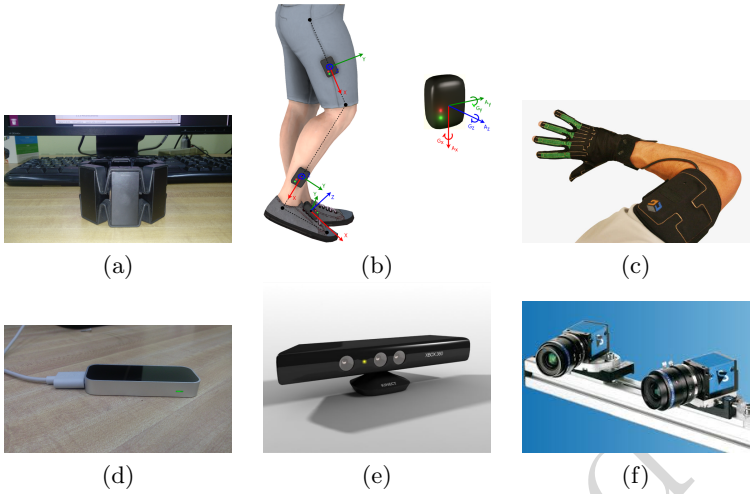

Figure 2. Vision and contact sensors: (a) *Myo arm band*; (b) inertial-based sensors; (c) *cyberglove* device; (d) *leap motion controller*; (e) *Kinect*; (f) stereoscopic camera array

2.2. Sensors for human gesture recognition

In the context of automatic gesture recognition, a sensor is defined as an electronic device that is able to converting an external physical signal into useful information to be sent toward a computer-human interface for high-level application development. In [5], the authors argued that two type of sensors have mainly been used: vision-based, and EMG wearable devices.

Multiple types of vision-based sensors have been used for these purposes; e.g., color and stereoscopic cameras, *Kinect* devices, and *leap motion controllers*. On the other hand, physical contact sensors offer different alternatives; some of the most common examples of these over the past few years are *Myo arm band*, *cyberglove*, and inertial-based sensors. Figure 2 shows some examples of commonly used body sensors for human gesture-recognition tasks.

Vision devices offer certain advantages over EMG (and vice versa); for this reason, it is best to merge the data that is collected from different sources through sensor-fusion techniques in order to improve the effective recognition of a particular gesture set. In this work, we propose the use of *leap motion controller* and *Myo arm band* devices to create large datasets that serve to characterize each gesture as being unique.

In Figure 2, some sensors that are used for gesture recognition are shown.

2.3. Predictive models

A predictive model is a computational technique that is formed from feature-extraction processes. This is used to predict trends and behavioral patterns and can be applied to multiple unknown events. Predictive analysis is based on the identification

of relationships between variables in past events; it then applies these relationships and predicts possible outcomes in future situations.

For human gesture-recognition topics, researchers have proposed methods that use different computational predictive models that apply machine-learning strategies. Wang [25] proposed a hidden Markov model implementation in order to recognize up to six human gestures using only EMG arm band sensors. The advantage of this is its capability of integrating temporal information to the inference process and classifying the recorded features during the tests for different hand gestures. Other authors have implemented feed-forward neural network architectures because of their knowledge-abstraction abilities and efficacy for regression tasks [4].

In [24], Sun et al. implemented Bayesian regression models in order to infer a set of arm-made gestures to gain good precision rates per the analyzed groups when many descriptors is used for each sample. Also, a finite state machine (FSM) consisting of computational abstractions that described the behavior of a reactive system through a specific number of states and a certain number of transition modelings was applied in [18].

According to our research, we have determined that the best models that match the purpose of automatically evaluating gestures that merge different sources of information are hidden Markov models and artificial neural networks. These approaches are detailed in the following sections.

2.3.1. Hidden Markov models

HMM is a technique that is widely used in signal processing. The essence of this technique is to construct a model that explains the occurrence of observations (defined as symbols) and use this to identify future observation sequences. The basis of HMMs and their applications were originally described in [20] and recently retaken for multiple tasks. For a hidden Markov model, there is a finite number of states; these is always in one of these. At each time, it enters a new state based on a transition probability distribution that is dependent on the previous steps. After a transition is made, an output symbol is generated based on a probability distribution that is dependent on the current state. Formally, an HMM is defined by the states, the transition probabilities among them, and the probabilities of the outputs given a state. For instance, the **Baum-Welch** algorithm is available to compute these probabilities and solve the problem of gesture recognition under this approach (according to [13]).

In this context, compute the probability $P(O|\lambda)$ of the occurrence for observation sequence $O = O_1, O_2, ..., O_T$ given that the λ model parameters can be solved by applying the **forward-backward** algorithm.

Selecting the best state sequence $I = i_1, i_2, ..., i_T$ so that $P(O, I|\lambda)$ is maximized can be addressed by the **Viterbi** algorithm. In addition to the λ learn model parameters given O such that $P(O|\lambda)$ is maximized, it is necessary the procedure **Baum-Welch**.

For gesture recognition, we assume that there are M different human gestures and N descriptors per sample. For the $j-$th feature ($j = 1, ..., N$), we learn one HMM for each class as well as the corresponding $\lambda_i, i = 1, ..., M$ parameters. Given one observation sequence O, we compute $P(O|\lambda)$ for each HMM by using the **forward-backward** procedure. Human gesture classification can be solved by finding the class i that has a maximum value of $P(O|\lambda_i)$ (as shown in Equation 1):

$$\text{gesture}(O) = \arg\max Ai : i = 1, ..., M(P(O|\lambda_i)). \tag{1}$$

We consider the set of these HMMs and the decision rule in Expression 1 to be a classifier for feature j. These M gesture classes and N features form an $M \times N$ matrix of HMMs. By HMM_i, j, we denote the model of gesture i (the i-th row) and feature j (the j-th column) and its corresponding parameters is $\lambda_{i,j}$. For the set of HMMs, row i is related to the gesture of interest. For these implementation of the HMM classifier, the following parameters are learned by the **Baum-Welch** algorithm:

1. prior probabilities of each state;
2. transition probabilities between states;
3. parameters of each state s, mean vector $\mu_{s,m}$, covariance matrix $\Sigma_{s,m}$, and weight $w_{s,m}$ of each mixture Gaussian component.

Both the **Forward** and **Baum-Welch** algorithms need to compute $P(O_t|s_t = s)$, the probability of observing O_t given that state s at time t. In practice, log-likelihood $log(P(O_t|s_t = s))$ is used instead. Another consideration is that the probability of the occurrence of observation sequence $O_1, O_2, ..., O_T$ tends to decrease exponentially as T increases. However, this causes no problem because the probability that is computed in each part of HMM_i, j ($i = 1, ..., M$) decreases comparably for feature set j.

2.3.2. Feed-forward neural networks

Artificial neural networks (ANN) are mathematical models that are constructed based on the functioning of biological neural networks. From a computational point of view, an ANN can be described as a set of cellular software pieces that are analogous to neurons, whereby a flow of information is established through an interconnection topology in the same way that a synapse works with biological cells. Some of the most important features that are described in this work are listed as follows:

- learn through examples;
- adaptability;
- generalization capabilities;
- fault tolerant.

There is at least one learning algorithm that is associated with each type of network. These consist of a systematic method for finding adequate values of the weights. In general terms, these are based on defining an objective function implicitly or explicitly (which represents the overall state of the network). From this, the initially

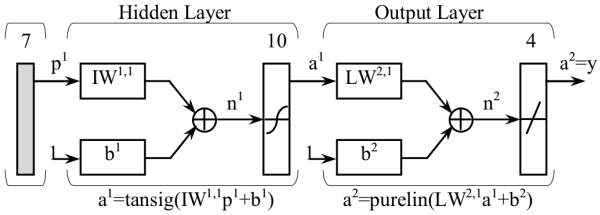

Figure 3. Feed-forward artificial neural network for gesture recognition proposed in [2]

assigned weights evolve to values that bring this function to a minimum stable state network. Therefore, learning is something that is characteristic of the type of network.

Here, the input vectors and corresponding target values are used to train the ANN until it can approximate a function. The designed ANN consists of three layers: an input layer, a tan-sigmoid hidden layer, and a linear output layer. For the second and third layers, weight matrix W, bias vector b, and output vector a are considered. Those weight matrices that are connected to inputs are called input weights, those that come from the hidden layer outputs are called layer weights. A representation of a whole neural network for this purpose is depicted in Figure 3.

The BP algorithm determines how to adjust the weights to minimize the performance by using the gradient of the performance function. The **Levenberg-Marquardt** algorithm is used for the training stage. At the end, the authors reported a final accuracy value of 90.3% for the four EMG gestures during the test stage.

In this work, we take both proposals (feed-forward neural networks and hidden Markov models) besides stages for data combination using the *leap motion controller* and *Myo arm band* for the input and training of the predictive models in order to increase the number of hand gesture groups as well as the accuracy rate values.

2.4. Data fusion

Data fusion from multiples sources is a compendium of multidisciplinary techniques that are analogous to the cognitive process that humans perform in order to integrate data from multiples sensors (senses) in order to make inferences about the outside world – converging on a set of results (a reaction).

One of the main objectives of data fusion is to combine the information that is obtained from different sources for making better decisions, reducing imprecision and uncertainty, and increasing robustness.

2.4.1. Data fusion technique

D.L. Hall and S.A.H McMullen classified fusion techniques according to the mathematical logic that is used to incorporate uncertainty. Such uncertainty can be attributed to observations or to any conclusions that are reached. The most important ways to incorporate this are as follows:

Probabilistic

This is the mathematical logic model that has a more powerful theoretical basis that is based on classical probability theory. By determining the functions of the distribution of the probabilities and conditional conditions (through empirical methods or stochastically), these impose very restrictive hypotheses with little credibility in complex problems. Classical probability and Bayes laws are probabilistic-based fusion techniques.

Evidential

Evidential logic defines non-additive probabilities as a general notion for logical assumptions and probabilities. The idea is to augment the standard propositional logic by considering an operator that represents the state of the knowledge of all of the sentences; it is argued that this is the best information that is available to an analyst. Dempster-Shafer and generalized evidential theory are evidence-based fusion techniques.

Diffuse

Diffuse logics was born in an article by L.A. Zadeh (published in 1965) that was entitled Fuzzy Sets-. This concept appeared in response to bialent classical logic, which allowed for a mathematical representation of concepts or an imprecise set. Thus, it included itself in multi-valued logic and admitted various values such as possible truths. Fuzzy sets and some hybrid AI operators are fusion techniques with this logical basis.

3. Body gesture-recognition using sensor fusion

Sensors fusion is a technique that consists of combining information that comes from multiple sensors in order to obtain a greater number of characteristics to describe each gesture in this case. The use of vision and physical contact sensors eliminates the disadvantages that each sensor presents when working independently. After a revision of the multiple methods that were aimed at merging the data, we have opted for feature level approaches in order to combine these two data sources.

The first step in the development of the proposed approach in this work is related to the adaptation of a *leap motion controller* as well as a *Myo armband* device to collect human gesture data. For all of our development and experiments, we have used a *robotic operating system* (ROS) and software packages that are aimed at recording and manipulating the data for these sensors.

3.1. Human gesture modeling

A gesture-recognition system has an initial analysis process that seeks to extract and estimate the characteristics that are used in the classification or recognition. To achieve this goal, the Myo armband and leap motion sensors will work together. The modeling of the gestures depends on the context in which they will be used.

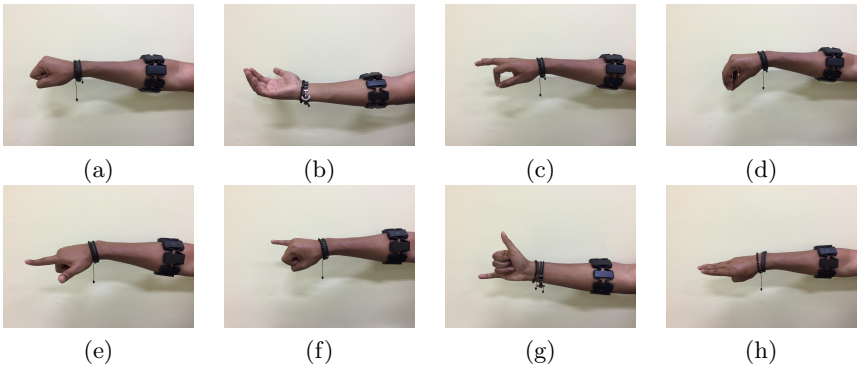

Figure 4. Proposed human gesture set: (a) giving; (b) needing; (c) grabbing; (d) holding; (e) setting free; (f) cutting; (g) drawing; (h) turning off

At first, the gesture set from [25] was taken to test the implemented predictive models in this work. A set of gestures was proposed that were aimed at enlarging the capabilities of our techniques in collaborative robotic environments. The proposed human gesture set is shown in Figure 4.

After defining the gestures of interest, we collected a large amount of data for each one through both the *leap motion controller* and *Myo arm band* in consideration of these multiple points of view, postures, and human users.

3.2. Implementation of predictive models

As described in Section 2.3, we implemented two predictive models: *feed-forward neural networks* (as shown in Section 2.4.1) and *hidden Markov models* (according to the approach that was described in Section 2.3.1). These two methods were modified to adapt each to the extracted features from our sensors.

3.2.1. ANN training

Training an artificial neural network is a process that modifies the value of the weights that are associated with each neuron.

For our training process, we used the **Levenberg-Marquardt** algorithm in combination with BP. These techniques are based on optimization methods for feed-forward neural architectures. We considered two different architectures for our tests (described as follows):

1. FF-ANN consisting of 3 layers: 8 neurons for input layer, 10 hidden neurons for middle layer, and output layer with 6 or 8 neurons according to considered human gesture set;
2. FF-ANN consisting of 3 layers: 73 neurons for input layer, 10 hidden neurons, followed by 6 or 8 output neurons.

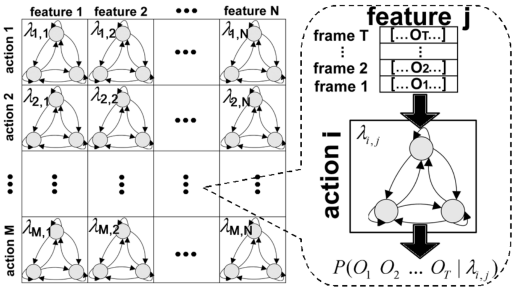

Figure 5. HMM matrix for gesture recognition [1]

The first one is considered when only the *Myo arm band* sensor is used to compute a signal vector of eight values from the muscle lectures of a human arm. The second architecture is considered when feature-level sensor fusion is carried out; then, we computed a signal vector of 73 values using the *leap motion controller* in addition.

During the training and testing stages, the collected dataset was split by following a ten-cross validation policy in all cases; in addition, two early-stopping conditions were used: a total mean squared error of $\epsilon \leq 0.001$, or the training is stopped after 1000 epochs.

The implemented neural network has eight signals in the input layer, has 10 neurons for training in the hidden layer and has six neurons in the output layer. The eight input signals correspond to the Myo armband readings, and the six neurons in the output layer represent each of the gestures.

3.2.2. HMM training

Equation $\lambda = \{N, A, B, \pi\}$ consists of four parameters; these elements will serve to perform the training of the models.

As described in Section 2.3.1, the **Baum-Welch** algorithm is used to train multiple HMMs by using the extracted features from the sensors. In the same way as described before, two kinds of models were implemented: using only the *Myo arm band*, and including the *leap motion controller* as well.

According to what is described in Section 2.3.1, we implemented an HMM matrix model for gesture recognition that was composed of six and eight individual HMMs according to the considered gesture set. Each HMM had five hidden states, and each state contained a 3-component mixture of Gaussian. Each $\lambda_{i,j}$ of $\text{HMM}_{i,j}$ is learned, and the probability of observation sequence $O_1, O_2, ..., O_T$ is computed. The gesture with the maximum probability in the same column is then selected. Figure 5 shows the proposed HMM matrix for gesture recognition considering two cases in the same manner as before: 8-values for the vector training, and 73-values for the fusion case.

4. Experiments and results

The following section describes all of the experiments that were performed for the recognition of the body gestures.

For our tests, we collected two datasets that were related to two different gesture types. This means that we had a total of 120,000 samples for *Dataset 1*. At the same way, *Dataset 2* was composed of eight human gestures according to Figure 4 in Section 3.1; it was composed of 5000 samples per gesture by four different human users, resulting in a total of 160,000 samples.

For all of the tests, we named our human users H_1, H_2, H_3, and H_4 and defined three training and evaluation cases for both the HMMs and ANNs as follows:

1. First Case:
   - ***Training:*** H_1; ***Testing:*** H_2.
   - ***Training:*** H_1; ***Testing:*** H_3.
   - ***Training:*** H_1; ***Testing:*** H_4.
2. Second Case:
   - ***Training:*** H_2; ***Testing:*** $H_1 + H_3 + H_4$.
3. Third Case:
   - ***Training:*** $H_1 + H_2 + H_3 + H_4$; ***Testing:*** $H_1 + H_2 + H_3 + H_4$.

These combinations were planned in order to analyze the ability of the models to represent each gesture independently from the human user and infer it from a different one. All of the experiments where performed following a 10-cross validation strategy. Other combinations were also considered, as we only presented the results for these cases.

4.1. Human gesture recognition using EMG

Our first objective was to evaluate the performance of the implemented predictive models by considering multiple training cases. During this test, we used *Dataset 1* while only considering EMG data in order to study the recognition rate with few features (one 8-feature vector per sample). Table 6 shows the precision and accuracy results for the feed-forward neural networks.

Table 2
HMMs recognition rate for *Dataset 1* while considering EMG signals

Gesture	**Case 1(a)**	**Case 1(b)**	**Case 1(c)**	**Case 2**	**Case 3**
To Stop	84.9%	7.2%	28.3%	87.8%	85.4%
To Continue	76.6%	8.5%	80.1%	81.9%	72.5%
Slow down	27.5%	1.7%	81.2%	75.0%	65.4%
Speed up	47.1%	8.8%	1.55%	45.6%	30.9%
To Need	24.5%	76.1%	3.7%	58.5%	60.1%
To Give	97.7%	99.9%	79.0%	95.5%	98.0%
Accuracy	**59.71%**	**32.49%**	**45.63%**	**74.05%**	**68.72%**

Table 3
Accuracy results for ANN and HMM

	ANN	HMM
Case 1	48.78%	45.94%
Case 2	56.18%	74.05%
Case 3	65.24%	68.72%

Table 1
ANN recognition rate for *Dataset 1* while considering EMG signals

Gesture	**Case 1(a)**	**Case 1(b)**	**Case 1(c)**	**Case 2**	**Case 3**
To Stop	29.8%	6.1%	60.4%	27.6%	70.4%
To Continue	71.6%	59.8%	96.8%	79.9%	89.3%
Slow down	91.4%	83.5%	81.3%	79.1%	87.3%
Speed up	21.5%	28.5%	51.6%	14.9%	47.6%
To Need	91.2%	9.6%	0.3%	70.4%	63.1%
To Give	59.2%	15.9%	19.7%	65.2%	33.7%
Accuracy	**60.79%**	**33.89%**	**51.67%**	**56.18%**	**65.24%**

The results shown in each column indicate the precision rate for each gesture while considering three cases. The last row of the table shows the accuracy recognition rate of each case. At the same way, these experiments were replicated using the HMM architecture as described before. These results are detailed in Table 2.

Table 3 summarizes the accuracy results of both predictive models. Showing these, the HMM models had higher accuracy rates at this stage for the three cases.

According these experiments, the ANN's accuracy resulted as follows: 48.78% for **Case 1**, 56.18% for **Case 2**, and 65.24% for **Case 3**. On the other hand, the HMM's accuracy resulted in 45.94% for **Case 1**, 74.05% for **Case 2**, and 68.72% for **Case 3**. Showing these, the HMMs could generalize the considered human gestures in better way while using fewer features for the last two cases.

Table 4

HMM recognition rate for *Dataset 1* while considering EMG and *leap motion* signals

Gesture	**Case 1(a)**	**Case 1(b)**	**Case 1(c)**	**Case 2**	**Case 3**
To Stop	85.1%	31.1%	33.1%	89.1%	99.3%
To Continue	77.1%	44.4%	85.1%	83.4%	99.8%
Slow down	51.1%	62.5%	92.2%	76.6%	99.9%
Speed up	57.9%	50.9%	32.8%	47.2%	99.7%
To Need	44.4%	77.1%	48.2%	61.4%	99.9%
To Give	98.2%	98.4%	77.78%	94.6%	98.6%
Accuracy	**68.99%**	**60.75%**	**61.56%**	**75.42%**	**99.52%**

4.2. Sensor fusion gesture recognition

For this stage, we tested for the precision and accuracy rates while considering the full feature vector for each gesture sample of *Dataset 1*. This included a 73-value vector that included the information from the *leap motion controller* device.

During this test, we collected evidence of precision and accuracy improvements when using both sensors being applied to *Dataset 1* for all six human gesture cases.

4.3. Eight-human-gesture-dataset proposal

Considering the experiments that were carried out in the previous sections, it is concluded that the HMM together with the sensor fusion significantly improves the recognition rate for the six human gestures in *Dataset 1*.

As described before, we proposed a new dataset (detailed in Section 3.1). *Dataset 2* is composed of eight different gestures. The results from testing the HMMs over the new dataset are shown in Table 5.

Table 5

HMM recognition rate for *Dataset 1* while considering EMG and *leap motion* signals

Gesture	**Case 1(a)**	**Case 1(b)**	**Case 1(c)**	**Case 2**	**Case 3**
Giving	68.1%	29.6%	43.8%	78.6%	91.8%
To Need	72.5%	34.5%	39.9%	43.8%	93.0%
Grabbing	68.4%	54.1%	59.0%	78.3%	99.5%
Holding	73.5%	63.5%	63.6%	98.0%	96.4%
Setting Free	58.4%	64.1%	3.7%	98.7%	97.9%
Cutting	73.9%	81.5%	82.1%	84.8%	97.2%
Drawing	63.7%	53.6%	57.4%	90.9%	96.8%
Turn Off	61.4%	67.2%	66.1%	63.9%	87.8%
Accuracy	**67.49%**	**55.69%**	**59.50%**	**79.63%**	**95.05%**

According to these, we received accuracy rates for the different cases as follows: 60.89% for **Case 1**, 79.63% for **Case 2**, and 95.05% for **Case 3**.

Table 6
Recognition percentage with new set

Gesture	First experiment	Second experiment	Third experiment
Off	77.1%	33.9%	87.8%
Cut	87.9%	84.8%	97.2%
Break free	98.6	98.7%	97.9%
Need	95.7%	43.8%	93.0%
Grab	81.2%	68.3%	99.5%
Hold	98.4%	98.0%	96.4%
Take	98.4%	33.6%	91.8%
Draw	97.2%	90.9%	96.8%

For this reason, these techniques were applied to a different set of gestures. For this stage of the experiments, the sampling was performed again for each gesture, and the same amount of data was taken (this time with four test subjects). Three different experiments were performed; these results are shown in Table 6.

The first column show the results of the first experiment that consisted of dividing the information that was obtained by each of the test subjects using 2500 information points for the training and 2500 for the testing. For the second experiment, a set of 5000 data points were used for the training, and 15,000 were used for the testing. The last experiment consisted of using the combined information of all of the test subjects, using 10,000 data points for the training and another 10,000 for the testing.

5. Conclusions and future work

In this paper, we have proposed feature-level fusion techniques that are meant to recognize a set of human gestures using myography and vision-based sensors: the *leap motion controller* and *Myo arm band* devices. Considering this combination, we gathered features that corresponded to specific human gestures from the hands and arms.

Since the main objective was to obtain the highest percentage of recognition for each gesture, it was decided to fuse the information from both sensors to achieve this purpose. It is important to mention that, when testing each sensor separately, quite reliable results were obtained; however, the percentages increased as soon as the information that was collected by the other device was incorporated. Immediately after the data was merged, it was used to train the predictive models. Both artificial neural networks (ANNs) and hidden Markov models (HHMs) were shown to be able to recognize the gestures; however, it is worth mentioning that the HMMs were especially busy for pattern and signal recognition and, therefore, showed superiority over the neural networks in this case.

We collected a set of eight different gestures that can be applied to industrial and medical robotic tasks in real scenarios. The combination of visual and myography

features to describe them has given good results, as the *Myo arm band* can estimate the postures of the hands and arms while the vision sensor has no complete view of arms. At the same time, the EMG lectures can be ambiguous for some similar gestures, and the *leap motion controller* helped us distinguish the slight differences between the hands when it was necessary.

For this research, we had a special interest in determining which kind of machine-learning approach is best-fitted to represent human gestures when considering these sensors. According to our tests, we determined that sequence-based techniques such as hidden Markov models are capable of incorporating any previous data in order to determine up to eight proposed gestures with high accuracy rates as compared to the neural network methods.

Our evidence during the tests suggested that the used techniques are able to operate in real-time application environments. As part of our immediate future work, we propose the implementation of these techniques in real activities such as robotic arm prototypes for industrial and medical tasks. Besides the ability of gesture recognition, our sensors are able to get the spatial positions of the hands among other high-level descriptors that are usable for developing robust applications. In this sense, our future objective in to incorporate the sensor lectures and data to the process of learning through a video sequence from human manipulation activities such those that are shown in Figure 6. Figures 1a, 1b, and 1c show a sequence of activities that are performed directly by the human hand; during movements that are performed with both arms, any data on the posture, velocity, etc. is being captured. Figures 2a, 2b, and 2c show the same routine; however, it is performed by a pair of robotic arms this time. The information that is captured by the sensors can be useful for the learning process of an automation task.

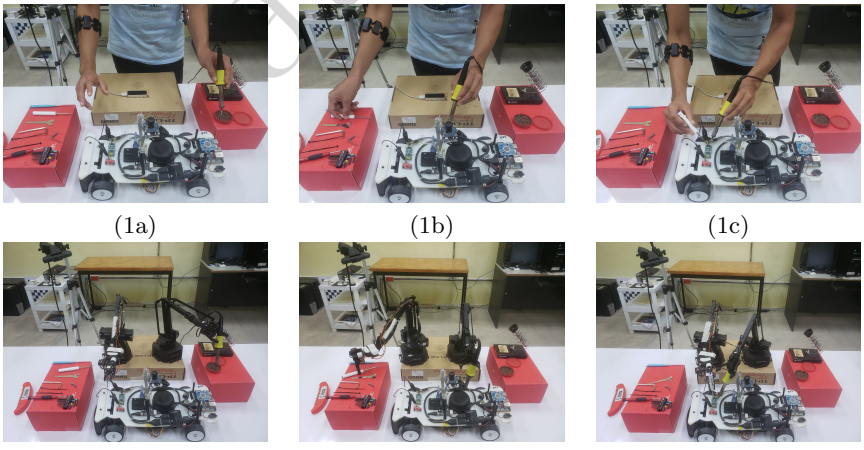

Figure 6. Human and robot activity sequence example – showing use of multiple tools for industrial applications

According to novel approaches such as [26], new techniques are emerging for improving machine-learning methods in robotic areas that are applied to industrial tasks. The developed work matches the objective of extracting information from human operators and developing remapping processes to transfer the knowledge to multiple robotic systems (as shown in 6). Currently, we are in the process of recording human and robotic sequences and developing techniques for carrying out these inferring parameter policy techniques between humans and robots by using sensors.

References

[1] Ahmad M., Lee S.W.: HMM-based human action recognition using multiview image sequences. In: *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1, pp. 263–266, IEEE, 2006.

[2] Ahsan M.R., Ibrahimy M.I., Khalifa O.O.: Electromygraphy (EMG) signal based hand gesture recognition using artificial neural network (ANN). In: *2011 4th International Conference on Mechatronics (ICOM)*, pp. 1–6, IEEE, 2011.

[3] Athavale S., Deshmukh M.: Dynamic Hand Gesture Recognition for Human Computer interaction; A Comparative Study, *International Journal of Engineering Research and General Science*, vol. 2(2), pp. 38–55, 2014.

[4] Benalcázar M.E., Anchundia C.E., Zea J.A., Zambrano P., Jaramillo A.G., Segura M.: Real-Time Hand Gesture Recognition Based on Artificial Feed-Forward Neural Networks and EMG. In: *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1492–1496, IEEE, 2018.

[5] FIGUEROA Y.S.: Reconocimiento anticipado de gestos, 2013.

[6] Georgi M., Amma C., Schultz T.: Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing. In: *Biosignals*, pp. 99–108, 2015.

[7] Jin H., Chen Q., Chen Z., Hu Y., Zhang J.: Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task, *CAAI Transactions on Intelligence Technology*, vol. 1(1), pp. 104–113, 2016.

[8] Jorgensen S.J.M., *et al.*: *Human detection, gesture recognition, and policy generation for human-aware robots*, Ph.D. thesis, 2017.

[9] Junker H., Amft O., Lukowicz P., Tröster G.: Gesture spotting with body-worn inertial sensors to detect user activities, *Pattern Recognition*, vol. 41(6), pp. 2010–2024, 2008.

[10] Keun-Cheol L., Kweon J.h., Kyung-Jae K., Choi J.w.: Method and apparatus for controlling a home device remotely in a home network system, 2018. US Patent 9,978,260.

[11] Kılıboz N.Ç., Güdükbay U.: A hand gesture recognition technique for human–computer interaction, *Journal of Visual Communication and Image Representation*, vol. 28, pp. 97–104, 2015.

[12] Lake S., Bailey M., Grant A.: Method and apparatus for analyzing capacitive EMG and IMU sensor signals for gesture control, 2016. US Patent 9,299,248.

[13] Lv F., Nevatia R.: Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In: *European conference on computer vision*, pp. 359–372, Springer, 2006.

[14] Mitra S., Acharya T.: Gesture recognition: A survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37(3), pp. 311–324, 2007.

[15] Molchanov P., Gupta S., Kim K., Pulli K.: Multi-sensor system for driver's hand-gesture recognition. In: *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, vol. 1, pp. 1–8, IEEE, 2015.

[16] Nagar P., Sengar A., Sarma M.: Hand shape based gesture recognition in hardware, *Archives of Applied Science Research*, vol. 5(3), pp. 261–269, 2013.

[17] Obaid M., Kistler F., Kasparavičiūtė G., Yantaç A.E., Fjeld M.: How would you gesture navigate a drone?: a user-centered approach to control a drone. In: *Proceedings of the 20th International Academic Mindtrek Conference*, pp. 113–121, ACM, 2016.

[18] Pink O., Becker J., Kammel S.: Automated driving on public roads: Experiences in real traffic, *it-Information Technology*, vol. 57(4), pp. 223–230, 2015.

[19] Premaratne P., Yang S., Vial P., Ifthikar Z.: Centroid tracking based dynamic hand gesture recognition using discrete hidden Markov models, *Neurocomputing*, vol. 228, pp. 79–83, 2017.

[20] Rabiner L.R., Juang B.H.: An introduction to hidden Markov models, *ieee assp magazine*, vol. 3(1), pp. 4–16, 1986.

[21] Rautaray S.S., Agrawal A.: Vision based hand gesture recognition for human computer interaction: a survey, *Artificial intelligence review*, vol. 43(1), pp. 1–54, 2015.

[22] Rautiainen T.T., Hui P., Kaunisto R.H.S., Teikari I.A., Ollikainen J.P.J.: Gesture control, 2016. US Patent 9,335,825.

[23] Sinha K., Kumari R., Priya A., Paul P.: A Computer Vision-Based Gesture Recognition Using Hidden Markov Model. In: *Innovations in Soft Computing and Information Technology*, pp. 55–67, Springer, 2019.

[24] Sun Y., Li C., Li G., Jiang G., Jiang D., Liu H., Zheng Z., Shu W.: Gesture recognition based on kinect and sEMG signal fusion, *Mobile Networks and Applications*, vol. 23(4), pp. 797–805, 2018.

[25] Wang W., Li R., Diekel Z.M., Chen Y., Zhang Z., Jia Y.: Controlling Object Hand-Over in Human–Robot Collaboration Via Natural Wearable Sensing, *IEEE Transactions on Human-Machine Systems*, vol. 49(1), pp. 59–71, 2018.

[26] Yu T., Finn C., Xie A., Dasari S., Zhang T., Abbeel P., Levine S.: One-shot imitation from observing humans via domain-adaptive meta-learning, *arXiv preprint arXiv:180201557*, 2018.

[27] Zabulis X., Baltzakis H., Argyros A.A.: Vision-Based Hand Gesture Recognition for Human-Computer Interaction., *The universal access handbook*, vol. 34, p. 30, 2009.

Affiliations

Domínguez Ramón Emmanuel
INAOE, emmanuel.dominguez@inaoep.mx

Díaz Hernández Raquel
INAOE,

Altamirano Robles Leopoldo
INAOE,

Received: 04.04.2020
Revised: 22.12.2021
Accepted: 23.12.2021