

Fuzzification of facial movements to generate human-machine interfaces in order to control robots by XMPP internet protocol

Jesús A. Romualdo Ramírez^{1,*}, Enrique Mendéz Franco^{1,**}, and David Tinoco Varela^{2,***}

¹UNAM Facultad de Estudios Superiores Cuautitlán, ITSE. San, Sebastián Xhala, Cuautitlán Izcalli Edo. De Mex. México

²UNAM Facultad de Estudios Superiores Cuautitlán, Departamento de Ingeniería. San Sebastián Xhala, Cuautitlán Izcalli.

Abstract. The classic human-machine interfaces require mechanical or electronic elements which can be cumbersome or complex in their uses and implementations. As a result, interfaces of such kind can present a rigid communication with devices which we want to control, additionally they may not be a usable tool by people who have lost a body limb and who present different types of corporal disabilities.

In this work is showed the development of a human-machine interface which can control a remote device by the characterization of natural human facial movements employing artificial vision and fuzzy logic.

Due to the characteristics of the proposed interface, it permits that disabled, untrained, and even illiterate persons can use it easily.

This implementation is able to establish a remote communication with any electronic device through the internet by the XMPP protocol, which gives it a dynamism of control over practically any geographical position in the world where internet connection exist, in this way, it is possible to integrate it into the internet of things.

Keywords - Human-machine interfaces, Fuzzy control systems, Artificial vision, Internet of things.

1 Introduction

Many human-machine interfaces are developed and used every day to control various electronic and mechanical devices, such as Smart TVs, drones and terrestrial, marine and even space reconnaissance robots. However, such kind of control needs of unnatural elements to work, for example: joysticks, radio frequency control (RF), teach pendants, and others.

This causes that a disabled or an untrained person can be confused and rigid when he tries to control any type of robot or technological device.

One of the biggest disadvantages of the classical interfaces is the fact that most of them are technological element which can be manipulated by hands, this presents an adversity when people who have lost their hands, or don not have mobility on them, tries to manipulate such interfaces.

Due to the above, the generation and developing of human-machine interfaces which respond naturally and which can be controlled by different types of corporal behaviors, such as facial expressions, permits that any person can operate any device intuitively. In fact, almost any invalid person can use this kind of devices, because the interfaces don not require a body extremity to manipulate these controllers.

Human body, and in a general form any biological entity, possess unique traits and biological signals which can be used to manage a robotic system in a natural, e intuitive way. Many of these signals can be given by the facial movements of any person.

When common expressions or daily movements from a human face are used as controllers, a robot can be more easily handled, due to there is no necessity to learn commands or sequence of instructions.

Nowadays with the emergence of the *internet of things*, and the connection of any technological device to the internet, there is a great importance to be able to

* gsusitto@gmail.com, ** itse.enrique@gmail.com, *** dativa19@hotmail.com

control robotic systems in big distances through a communication global network, for this reason a human-machine interface which can control and monitor any electronic device by the network of networks could has a great scientific and commercial impact. The advantages of this schema can increase with respect to the inclusion of population sectors with body disabilities, and population who cannot read or write. All of this, just because there is a more natural communication between the human and the device to control.

In the present paper, it is demonstrated that through the use of techniques of artificial vision it is possible to generate more ergonomic e intuitive controllers which can perform command actions in any kind of environments. The presented controller uses fuzzy logic to interpret the facial movements in order to convert them to control words.

It is important to mention, it is necessary that the system be of low cost, and it needs to have the basic infrastructure to exchange information in order to have a penetration in population sectors of different social levels. Due to this situation, a project which uses the internet communication protocol XMPP (*Extensible Messaging and Presence Protocol*) is proposed. XMPP was originally destined to send and to receive instant messaging. This protocol permits us, among other things, to transmit control signals between two separated identities, and to make easier their connection because anybody can use the free servers destined to this kind of tasks.

2 Preliminaries

2.1 Fuzzy logic

Fuzzy logic arises between the 60s and 70s, proposed by *Lofiti A Zadeh* [1]. This logic permits to simulate the mechanisms of human reasoning in order to control the proposed system. It provides a mathematical model which can process the uncertainty from the human cognitive process, and in this way it can be possible to solve different kind of problems using mathematical and computational methods.

Basically, fuzzy logic is a manner to mathematically represent the uncertainty, and vagueness of a problem. When we are looking for a solution of a given problem, we have input and output variables, fuzzy logic allows the transformation of the input variables into its corresponding output privileging the significance characteristics instead of the precision ones.

Fuzzy logic uses real values between 0 and 1 to indicate if a fact is completely false (0), or totally true (1), this is the membership degree corresponding to an input value in a fuzzy set. Being multivalued, an input can belong to more than one fuzzy set at the same moment, with a different membership value for each

one. In this way, it provides a smooth transition between a pair of characteristics. All values which can take an input variable are called *universe of discourse*.

Once we have determined the different membership values of an entry to each of the sets in the universe of discourse we say that the variable has been *fuzzified*.

A standard control system based in fuzzy logic is showed in figure 1.

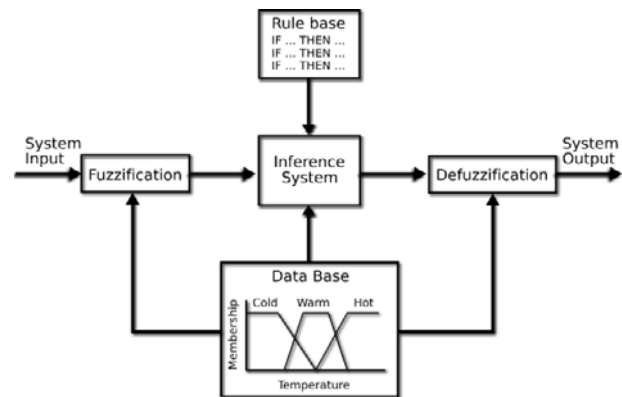


Fig. 1. Basic scheme of a fuzzy system.

2.2 Artificial vision and OpenCV

Artificial vision (AV) is a scientific discipline which consists in acquiring, processing, and mathematical modeling the processes of visual perception in living beings, capturing images from the real world in order to generate numerical or symbolic data that can be quantified and codified and then used that information in algorithms and software. Collected data can be used in the decision-making needed to carry out a given process. A system which uses artificial vision recognize figures and patterns, among other things.

Nowadays, artificial vision has great popularity over many technological, industrials, and services sectors due to the large amount of applications in which can be directed. AV can solve problems of supervision in fruits and vegetables [2], obstacles detection systems in vehicles and smart transportation [3, 4], and even it is usefulness in the design, development, and generation of systems which can be helpful as prosthesis for blind people [5] [6].

Due to the big amount of uses that this kind of systems can has, there is many computing tools developed in order to facilitate the work of new researchers and developers in this knowledge area.

The number of software instruments to work with these schemes is in fact varied, ranging from the costly to the free, and from the private to the open access, however, for the development of this project we had used the Open CV (*Open Source Computer Vision*) library due to it is no restricted just to one operative system, it is compatible with *Linux*, *Windows* and *Mac OS X*, it has

big dynamism, and additionally it has interfaces to work with Python^a, Ruby^b, Matlab^c, and other languages.

Open CV is an open source library developed by *Intel* to be used in C language. It is oriented to work with real-time applications, for this reason it is a really good option to use it in facial recognition [7], tracing movements and objects detecting tasks [8], and functionality on mobile devices [9]. The last of great importance and interest for the development of the proposed project.

2.3 XMPP, communication protocol

Extensible Messaging and Presence Protocol XMPP, is an open protocol based on XML (*eXtensible Markup Language*), designed to give solutions related to real-time communications. This protocol is daily used by many companies^d.

Some of the most notable advantages of XMPP are:

1. *Decentralization*, XMPP Works with decentralized client-server networks, due to this fact anyone can install their own XMPP server, and make a connection to the rest of the network using the internet infrastructure or by a subscription to a public server.
2. *Safe*, the protocol core has SASL (*Simple Authentication and Security Layer*) and TSL (*Transport Layer Security*), which provides encryption and safe authentication algorithms.
3. *Extensible*, due to the XMPP cores is based in XML, it is possible to obtain custom applications such as instant messaging services, voice over IP, communication between machines, notifications, and alerts. In order to maintain the interoperability of the XMPP applications, *software foundation* manages the most of the extensions.
4. *Open standard*, Internet Engineering Task Force (IETF)^e has formalized the XMPP protocol as an instant messaging standard [10].
5. XMPP protocols are free, public and easy to understand.

All the presented advantages of the XMPP protocol are useful in the development of human-machine interfaces which can control technological devices remotely.

2.3.1 Basics of XMPP

Each XMPP entity requires an identifier called a JID (*Jabber identifier*), which is composed of the user name,

^a <https://www.python.org/>

^b <https://www.ruby-lang.org/>

^c <https://www.mathworks.com/products/matlab.html>

^d <https://xmpp.org/uses/social.html>

^e <https://www.ietf.org/>

and the domain to which the resource is connected. The format of this identifier is like that of an email address "user @ domain / resource".

Using XML transmission technology, XML elements known as *stanzas* are exchanged between two entities in a network. A stanza can be understood as a basic unit of information. There are three types of stanzas in XMPP:

- a. The *message stanza*, transmits information from one user to another, contains information from the sender and recipient, as well as the content and subject of the message.
- b. The *Presence stanza* reports on the availability and status of a user on the network. Only those who have the contact of a user can see their status.
- c. The *IQ* (Information / Query) *stanza*, provides a request-response structure between network users.

2.3.2 Process of delivery of the messages.

Assuming two users with subscriptions in different domain, `quijote@lamancha.net` and `dulcinea@toboso.net`, when Quijote sends a message to Dulcinea the following events are triggered (see figure 2).

1. Quijote connects to lamancha.net server.
2. The server lamancha.net negotiates and opens a connection with toboso.net, in case that the connection is not achieved, the message is discarded.
3. The server toboso.net delivers the message to Dulcinea, if it is not online, the message will be delivered later.

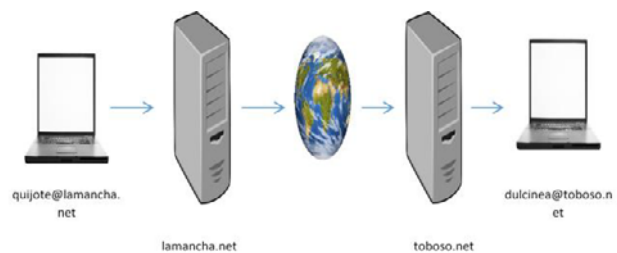


Fig. 2. Basic scheme of sending and receiving data through XMPP.

2.4 Internet of things

The *Internet of Things* (IoT) is a paradigm that quickly gains ground in the world of modern wireless telecommunications. The main idea of this paradigm is that many elements such as sensors, TV devices, appliances, telephones, among others, are able to interact between each of them through the internet [11].

In this area, XMPP can not only be used as a protocol to send instant messages between two humans, but can

be used to send commands between humans to objects, sensors or actuators and in this way, an automated remote control.

2.5 Raspberry Pi

Raspberry Pi^f is a reduced-board computer, meaning that all its components, including processor, memory and ports of entry and exit, are integrated into a single board. The processor that has the board is an ARM^g that runs at about 700 MHz, its RAM is 1 GB, and both in addition to the graphics processor are encapsulated in a single chip. Raspberry has user-configurable input and output pins that are programmable by the user, allowing you to interact with sensors and actuators. System storage is done on a micro SD card in which the operating system is also installed, this allows to change the functionalities of the card only by changing micro SD card. Raspberry can connect to the internet through an Ethernet port or in some models using WiFi, allowing you to interact with remote applications too.

The software that can be installed in Raspberry Pi as an operating system is commonly based on Linux distributions for ARM, such as Android, Firefox OS, Pidora but the most common and also officially supported is Raspbian^h, based on Debianⁱ, which has preinstalled features destined to the development of applications, like Python and Mathematica among others.

Raspberry Pi was developed by the *Raspberry Pi Foundation* in the UK, intended to bring young people to the development of skills in computer science, but the low cost, small size, ease of connection to the internet and interaction with sensors and actuators have expanded the uses that can be given to this computer. In this sense the applications can go from a personal computer, an entertainment center or a system dedicated to specific tasks in the scientific, academic or industrial sphere [12].

An important fact to use Raspberry Pi is that it has been analyzed as an element that can be linked quickly within the IoT network [13].

3 Development

A remote control system was developed by means of Internet protocol XMPP, the control commands are obtained by means of an artificial vision system and fuzzy logic that identifies the facial movements of a person.

The prototype is composed of two entities, one whose function is to generate the control words for the mobile

robot through artificial vision and a fuzzy inference process, this will be called «console». The second entity is responsible for receiving control words from the console and activating the actuators in the mobile robot. This entity will be called “mobile receptor”.

Communication is done using the internet as a transmission channel, using the XMPP protocol for sending data. This protocol is used in order to save resources by avoiding installing an own server and making development faster by taking advantage of existing infrastructure and internet services, the basic scheme of this system can be seen in figure 3.



Fig. 3. Basic scheme of the developed prototype.

3.1 Console

The console is the part of the prototype from where the mobile receiver is controlled. This entity is executed on a computing system complemented with an image receiving and capturing element such as a webcam.

The first step that the console performs is the acquisition of data through the camera, when recognizing that there is a face in its field of vision, the position of the center of the face is calculated, a value that is used as a reference point to determine the commands that control the mobile receiver.

One of the main features of the application is that once it has detected the face that will give the orders, the software follows the movement of the face within its particular field of vision, and this situation allows the operation to be performed in real time.

Once the system has defined that a particular facial movement has been a command and not a sporadic movement, the facial position is calculated and the control word is obtained, it is at this time when the control word is sent as a text message through the internet to the robot or element to be controlled, as can be seen in figure 4.

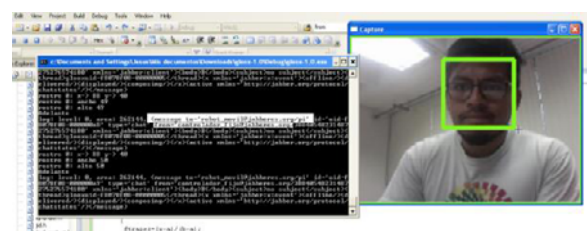


Fig. 4. Detection and sending of messages through XMPP to a remote robot.

^f <https://www.raspberrypi.org/>

^g <https://www.arm.com/>

^h <https://www.raspbian.org/>

ⁱ <https://www.debian.org/index.es.html>

To make the interface, the OpenCV library was used to generate facial recognition, and Gloop as an XMPP client.

The program follows the steps showed in Figure 5.

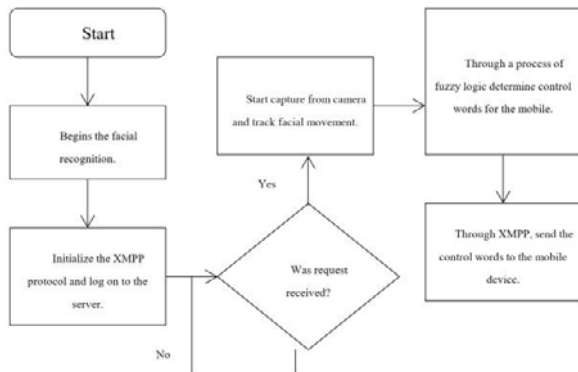


Fig. 5. Flow diagram corresponding to the behavior of the face detection system and sending control signals to a robot positioned in a different geographical area.

2.3.1 Creation of the instant messaging session.

When the console is run, it connects to the messaging server and creates a remote session in which it reports its availability online, waiting to receive a message to open a chat session with the mobile receiver. Once session is opened, it starts facial detection and the process of generating and transmitting control words to any remotely connected device.

2.3.2 Detection and facial tracking

The description of the process completed by the console to make the facial recognizing and tracing is listed below.

- 1) Capture the image from the camera.
- 2) Convert the image to grayscale.
- 3) Equalize the histogram of the grayscale image to standardize the brightness and contrast of the image, so that in different lighting conditions the image content is not affected.
- 4) Once the image is processed, the classifier to be processed is loaded.
- 5) When the face has been identified, a pair of XY coordinates are generated for the upper left and lower right corners, these coordinates serve to generate a recognition frame in the area where the face is located, and therefore, to calculate the center of this area, as can be seen in figure 6. The center coordinates will be used as inputs for the diffuse inference system.

3.2 Inference system

It was decided by fuzzy inputs as they allow a gradual transition between variables, also if the user is into the camera's viewing and he makes a subtle facial movement but without the intention of generating a change in the state in which the robot is, the fuzzy input allows a tolerance rage to avoid fake commands.

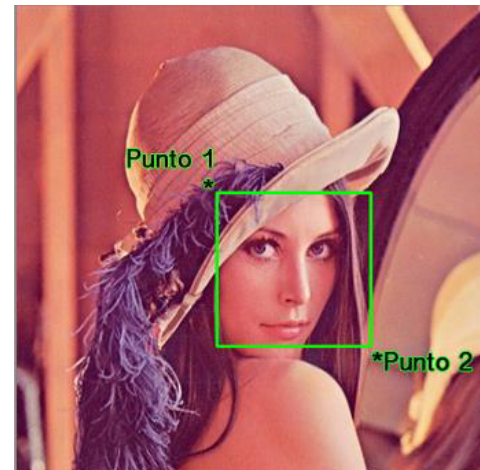


Fig. 6. Face identification by coordinates XY.

The camera's viewing area is divided into two axes, X, and Y, being the origin the upper left corner so that the values of the X axis grow to the right and those of the Y axis to the left both in a positive way.

Each axis as outputs universe of discourse is divided into three membership functions which can be observed in figure 7.

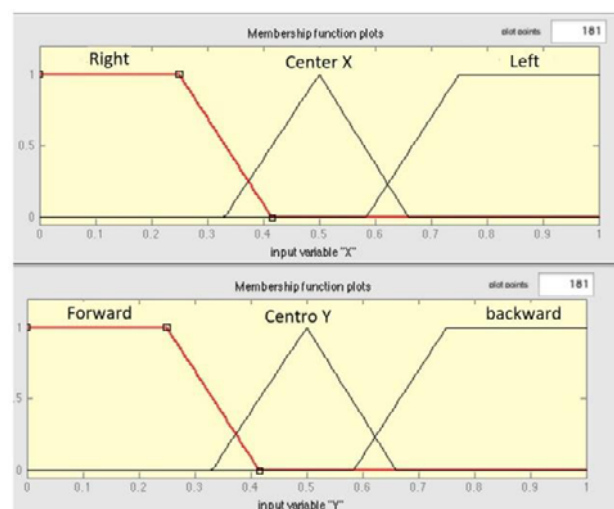


Fig. 7. Membership functions used for the identification of facial movements.

For the inference system was developed a sugeno-type fuzzy inference with just one output variable, the variable has nine sets with constant value. With the inputs, and outputs sets were generated the rules showed in the figure 8.

With the rules showed in figure 8, it has been generated 9 different control words which will be sent to the mobile receptor. These words indicate to the robot the movement it needs to do. There is a rule classifier with the velocities and directions programmed from the beginning in the mobile receptor.

	Left	Center X	Right
Forward	Forward Left	Forward	Forward Right
Center Y	Left	Stop	Right
Backward	Backward Left	Backward	Backward Right

Fig. 8. Used rules in the inference system.

The inference system can be modified in order to have two continues output sets, a set by each actuator. Each set has to transmit to the actuators, or CD motors, the gradualness in the velocities and directions of the controlled robot.

When the inference process has generated the output command, this is passed as argument to the method which send it through the messaging session that has been created previously. When it is ended the sending of the command, the program continues in a loop generating new commands until the session is finalized.

The session ends when the robot sends a message indicating to the console "terminate the connection".

3.2 Mobile receptor

The mobile receptor is formed by the unit of control, power stage, and actuators.

Unit of control is formed by the Raspberry Pi board which receive, process, and manage the commands that have been sent through the internet. Raspberry uses the processed data to activate the actuators, cameras or any device connected to the robot, and controller them according to the received instructions.

It is very important to say that the Raspberry PI GPIO pines are unbuffered and it has the level voltage of 0 volts for the logic zero, and 3.3 volt for the logic one, limited to 500mA. These levels are good enough to connect LED's (Light Emitting Diode), but they can destroy the board if a motor is directly connected, for that reason it is necessary to have a power stage between the board and the actuators. The power stage in this prototype is made with a simple electronic device: the H

bridge^j which generates the necessary energy to turn on the actuators.

In the proposed model, Raspberry was loaded with Raspbian^k as operating system because it has development tools such as a Python^l interpreter.

The mobile receptor is ready to start a chat session and information exchange with the console when it has been successfully connected to the jabberes.org server, these steps can be seen in the figures 9, and 10.

```

pi@raspberrypi ~$
File Edit Tabs Help
DEBUG: dispatcher ok <iq id='1' type='result'>
<bind xmlns='urn:ietf:params:xml:ns:mpp:bind'
<id=robot_mov.id@jabberes.org/pi/>/jid'
</bind>
</iq>
DEBUG: dispatcher ok Get jabber:client:in stanza
DEBUG: dispatcher ok Dispatching in stanza with type=result prepq=1/1 id=1
DEBUG: dispatcher ok Expected stanza arrived
DEBUG: bind ok Successfully bound robot_mov.id@jabberes.org/pi.
DEBUG: socket sent <iq type='set' id='2'>
<session xmlns='urn:ietf:params:xml:ns:mpp:session' />
</iq>
DEBUG: dispatcher wait Waiting for ID:2 with timeout 25...
DEBUG: socket get <iq type='result' id='2'>
DEBUG: dispatcher ok Get jabber:client:in stanza
DEBUG: dispatcher ok Dispatching iq stanza with type=result prepq=1/1 id=1
DEBUG: dispatcher ok Expected stanza arrived
DEBUG: bind ok Successfully opened session.
DEBUG: dispatcher info Registering handler <bound method Bot.mpp.message of
Bot instance at 0x7c36c0b0 for "message" type>: mpp:client:

```

Fig. 9. Connection between the console and the Raspberry Pi installed in the remote robot to control.

```

pi@raspberrypi ~$
File Edit Tabs Help
DEBUG: client: p jabber:client: u http://jabber.org/protocol/chatstates' id=
id=robot_mov.id@jabberes.org/pi/>
</chatstate>
</chatstate>
</iq>
DEBUG: socket sent <message from='controller_f1@jabberes.org' id=388405482
31487272570574180 to='robot_mov.id@jabberes.org/pi' xml:lang='en' id='uid-180781
08-00000008' type='control'>
<body></body>
<body></body>
<body></body>
<body></body>
<body></body>
<body></body>
<body></body>
<body></body>
<body></body>
</message>
</iq>
DEBUG: dispatcher ok Get jabber:client:message stanza
DEBUG: dispatcher ok Dispatching message stanza with type=chat prepq=1/1 id=
id=robot_mov.id@jabberes.org/pi/>
</chatstate>
</chatstate>
</iq>

```

Fig. 10. Reception and interpretation of control commands, execution made by the remote Raspberry Pi.

4 Results

From integrating techniques and methods described through the paper was achieved:

1. It was established a control system using artificial vision. In this, fuzzy logic allows generating command words through the interpretation of facial movements.
2. It was made a connection between two identities using a free real-time communication protocol

^j An H bridge is an electronic circuit which allows that a DC electric motor can rotate in both ways forward and backward.

^k <https://www.raspbian.org/>

^l <https://www.python.org/>

through the internet, sending and receiving command words.

3. It was made a remote control of a robot using a computing reduced system as interface to generate access to the internet.
4. We could control a robot in a different geographical position by just making face movements, the robot started, moved to different directions, and turned off. It could be possible to monitoring the environment in which it was moving.
5. The generated system can easily connect to the internet of things, and so, interact with any connected element.
6. It was developed a control system easy and intuitive for final users. It can be used for people with corporal disabilities, untrained, and even can be used by people who cannot read and write. This is a very important fact due to this system is easily incorporated in any population sector.
7. It was developed a low cost system, and for that, accessible to population sectors with low economic resources.

5 Conclusions

Artificial vision as means of capture of the input variables in a control system gives to users a different experience in which there is no necessity to use only fingers or hands as control elements, but they involve a different mobility for its execution.

Fuzzy logic permitted processing efficiently the information obtained by the camera, mainly because this logic is more tolerant to errors than classical logic, it allows an adequate and flexible control in reference with the image capture, and so a better recognition of command words. By means of fuzzy logic, it is possible to avoid mistakes when the control words are generated because the proposed system can distinguish between movements that indicate a command and those movements that just are generated by a reaction not related to the control of the mobile receptor.

The XMPP instant messaging protocol as means to send commands through internet allowed to facilitate the process. The fact that it is possible to use the free servers represents a considerable saving in the economic and in the temporal. It is not necessary to have an own server, and it is not necessary to waste time in develop an information exchange protocol.

The use of a reduced computer such as Raspberry Pi facilitates the development of prototypes which requires of functionalities more complex in comparison with a microcontroller which has not many resources. Its multiple forms of connection, and its integration with standard peripherals via USB or low-level hardware make it ideal to interact with the real world through sensors and actuators. The easy form to connect it to

internet positions it as a suitable device to control different technological, electronic, and mechanical systems remotely.

One of the most interesting advantages of the proposed system is the fact that it allows an easy, and fast interaction with the internet of things, due to it can make hardware-hardware communications in a manual or autonomous way.

The most important advantage of the proposed model with respect to classical human-machine interfaces is that it can be easily used by people with corporal disabilities, even it can be used by illiterate people because this system does not need complex instructions or previous technical knowledge, it is not necessary to include hands, fingers, foots or any corporal extremity to control it, the system is entirely controlled just by natural facial expressions. All of this allows that almost anyone can use it. By means of the inputs data fuzzification, it is possible to particularize the gestures that can be used as control words for anyone.

It is important to say that the development of this project was made by using free use hardware and software.

Acknowledgment

The authors thank PAPIIT IN113316 and the program PI-API1634 of the FESC-UNAM.

References

1. Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, **8**(3), 338-353.
2. Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., & Blasco, J. (2011). Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food and Bioprocess Technology*, **4**(4), 487-504.
3. Bertozzi, M., Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., & Porta, M. (2002). Artificial vision in road vehicles. *Proceedings of the IEEE*, **90**(7), 1258-1271.
4. Bertozzi, M., Broggi, A., & Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous systems*, **32**(1), 1-16.
5. Dobelle, W. H. (2000). Artificial vision for the blind by connecting a television camera to the visual cortex. *ASAIJ journal*, **46**(1), 3-9.
6. Chader, G. J., Weiland, J., & Humayun, M. S. (2009). Artificial vision: needs, functioning, and testing of a retinal electronic prosthesis. *Progress in brain research*, **175**, 317-332.
7. Youssif, A. A., & Asker, W. A. (2011). Automatic facial expression recognition system based on geometric and appearance features. *Computer and Information Science*, **4**(2), 115.

8. Uke, N., & Thool, R. (2013). Moving vehicle detection for measuring traffic count using opencv. *Journal of Automation and Control Engineering*, **1**(4).
9. Pulli, K., Baksheev, A., Korniyakov, K., & Eruhimov, V. (2012). Real-time computer vision with OpenCV. *Communications of the ACM*, **55**(6), 61-69.
10. Saint-Andre, P. (2011). Extensible messaging and presence protocol (XMPP): Core.
11. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, **54**(15), 2787-2805.
12. Upton, E., & Halfacree, G. (2014). *Raspberry Pi user guide*. John Wiley & Sons.
13. Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of things hardware: performances and constraints. *Design issues*, **3**, 8.
14. Hjeltnæs, E., & Low, B. K. (2001). Face detection: A survey. *Computer vision and image understanding*, **83**(3), 236-274.