ISSN 1870-4069

Automatic Translation System from Mexican Sign Language to Text

Luis Alberto Flores Montaño, Rosa María Rodríguez-Aguilar

Universidad Autónoma del Estado de México, Unidad Académica Profesional Nezahualcóyotl, Departamento de Ingeniería en Sistemas Inteligentes, Estado de México, Mexico

United_fenrir@hotmail.com, rmrodrigueza@uaemex.mx

Abstract. The Mexican Sign Language (*Lenguaje de Señas Mexicano*, *LSM*) consists of movements of the human body and hands to communicate, this is used by the deaf-mute to articulate thoughts and emotions. Thus emerges the initiative to device an Automatic Computer System for the Translation of Sign Language to Text, in order to facilitate communication to deaf and non-deaf people. The system is based on pattern recognition. This recognition is applied continuously using a capture device, in this case a Web camera using the Visual Studio IDE specifically with the OpenCV library. The recognition of the signs is made by comparing a matrix, previously obtained by means of several algorithms for the extraction of the object of interest, in this case the hand. In tests for the recognition of numbers, an average reliability percentage of 80%was obtained for the patterns analyzed.

Keywords: Pattern recognition, Mexican sign language, image processing, segmentation.

1 Introduction

Human communication is the transmission of information through various languages, among which, oral expression is the ideal way to share ideas, thoughts, emotions, etc. Human communication has as its main purpose that the individual achieves a satisfactory social integration that (in turn) allows the scientific and technological development that society requires.

However, hearing and visual disabilities of people is a problem that continues to demand immediate attention because such disabilities limit the opportunities for social, emotional, educational, professional and cultural integration that every human being deserves. People with visual impairment find support in a system of engraved symbols (Braille language) registered by means of touch for reading and writing. The hearing impaired people use sign language (signs articulated with movements of the hands and fingers) to communicate. In particular, sign language uses various movements generated with the body (corporal movement) and the face (gestural movement) to emphasize the expressions established by the hands and fingers, thus allowing to establish a communication that, despite being non-verbal, is significant.

Currently, the rapid development of intelligent computer systems has begun to impact positively on the social integration of the auditory disabled with applications that facilitate their non-verbal communication. For the recognition of a gestural movement, there have been a series of works based on the digital processing of signs, [which are] captured in video-images in order to be translated into text with the possibility of voice assigning for mobile devices. Such applications recognize body and gestural movements in real time by means of techniques such as neural networks for non-supervised learning [1].

1.1 Mexican Sign Language

Sign language is considered one of the earliest forms of communication used by the deaf community, [and it] consists of a series of gestural signs articulated with the hands, accompanied by facial expressions, intentional look and body movements. In 1620, Juan Pablo de Bonet in Spain, wrote the first book to teach sign language for deaf-mute people. In 1755 Charles Michele L'Epee in Paris, France founded the first school for people with hearing disabilities in the world. In 1778 Samuel Heinicke of Leipzig, founded the first school for deaf-mutes in Germany. In the United States the natives had a communication system similar to that of deaf-mute people [2].

In Mexico, the Mexican Sign Language (*Lenguaje de Señas Mexicana, LSM*), which is composed of the dactylology and ideograms, is used. Where dactylology is sign language generated by the gestures of the hand, which seek to interpret each letter of the alphabet through different shapes or forms, as shown in Figure 1. On the other hand, the ideograms are representations of words with one or more hand configurations [2].

Fig. 1. Mexican Sign Language (LSM). [3]

El Colegio de México [The College of Mexico] developed the Mexican Spanish-Sign Language Dictionary (*DIELSEME*) to support the purpose of contributing to socialization with the *LSM*, since it is a useful tool for teaching and learning the language.

The *DIELSEME* is considered bilingual because it is an answer to the basic need to teach the *LSM,* with reference to written Spanish, to deaf people as well as to a variety of potential users: [*i.e.*] hearing parents with deaf children, teachers of all educational levels teaching deaf students, interpreters in training, interpreter schools and society in general [3].

2 Used Algorithms

The recognition of the signs is made by comparing a previously obtained matrix, — by means of the Pavlidis algorithms (for contour), the three coins algorithm (for convex hull), the Skyum algorithm (for minimum enclosing circle), and lastly the Wen and Niu algorithm (for the detection of fingertips) — for the extraction of the object of interest, in this case the hand.

2.1 Theo Pavlidis Algorithm

The Theo Pavlidis Algorithm is used to find the outline of an image. [4] This algorithm begins with a starting point; having located this starting point, the scanning of each row of pixels can be executed, starting from the lower left-hand corner. The scanning of each row starts from left to right. When the white pixel is found, it is declared as the initial pixel. After identifying the initial pixel, three more pixels, P1, P2 and P3 — which are of interest for the algorithm [5] — are located.

P1 =Upper left pixel
P2 =Upper pixel
P3 =Upper right pixel

All points are relative to the start pixel, as shown in Figure 2.

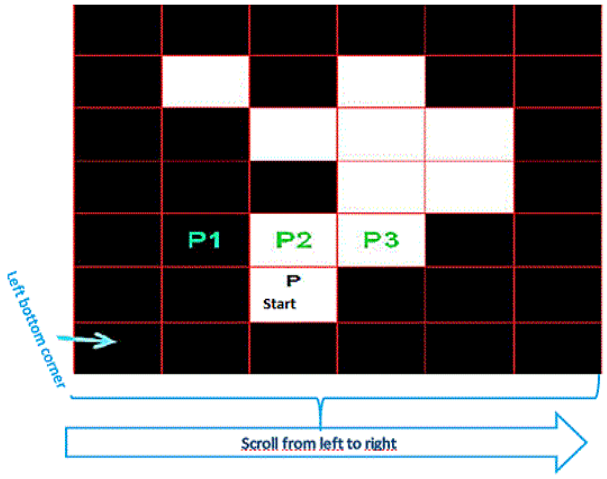

Fig. 2. Algorithm of Theo Pavlidis, position of P1, P2 and P3.

When the start pixel is located, there are 4 [possible] cases to consider for the recognition of the contour of the image, they are enumerated in detail in the following algorithm:

Input: *A binary image that contains a group of white pixels.*
Output: *A sequence with B (b1, b2… bk) of the limit pixels as contour.*

Start
Scan each pixel in rows from the lower left part until a white pixel is found s
Insert s in B and adjust so that it is a START pixel.
while (s has not been visited 2 times)

If pixel P1 is white
─Insert P1 in B
─Adjust s = P1
─Move one position forward followed by a new position in the current [one] to the left
else if P2 is white
─Insert P2 = B
─Adjust s = P2
─Move one position forward
else if P3 is white
─Insert P3 in B
─Adjust s = P3
─Move one position to the right and actualize its position, move one position in the current [one] to the left
else if s has been rotated 3 times
─End process and declare s as an isolated pixel.
else
─Rotate 90° clockwise while P is positioned over the current pixel
End

Fig. 3. Formal description of Pavlidis Algorithm [5].

2.2 Convex Hull and Convexity Defects

The convex hull method or three-coin algorithm is an efficient and robust procedure to obtain the structural parts of the hand, which consists in calculating a convex envelope of the contour of the hand and [then] comparing it. The calculation of a convex polygon envelope consists in obtaining a polygon, the sides of which connect the outermost points of the contour, thus eliminating any concavities or convexity defects that might arise [5].

Obtaining the convexity defects allows the identification of the interdigital spaces, in addition to other concavities generated by the shape of the hand or errors in calculating the contour. The defects are represented with their starting point, their end

point and their point of maximum depth [5]. See figure 4 for details of the convex hull algorithm.

Input: *A set of N-points that form a polygon on a plane.*
Output: *A set of points that form the convex hull of the input.*

Start:

- *Find the point located furthest to the left as a start point. Label it as p0.*
- *Label the rest of N-1 points in a clockwise direction.*
- *Place the three coins on p0, p1, p2 and label them respectively as end, middle and front coin.*

Make up to: "the front coin", is the start point, and from there rotate to the right. if the three coins form a right hand rotation

- *Take "end coin", place it on the next point following "front point".*
- *"End coin" is now "front coin", "front coin " is now "middle coin", la "middle coin" is now "end coin"*

else coins make a left hand rotation.

- *Discard the point (associated to the borders) that is on the "middle coin".*
- *Take the "middle coin", and place it on the former "end point".*
- *"Middle coin" becomes "end coin", "end coin" becomes "middle coin".*

End
The remaining points form a convex hull when connected in an order.

Fig. 4. Formal description of the three coin algorithm.

2.3 Minimum Enclosing Circle

The minimum enclosing circle is used in computational geometry, sometimes in planning to locate a shared facility in an easy way. A hospital or a post office are good examples of a shared facility. If one considers each household in the community as a point in the plane, the minimum center of the enclosing circle is a suitable location for [the] shared installation, as shown in figure 5. This idea is applied to find the position of the hand palm [6].

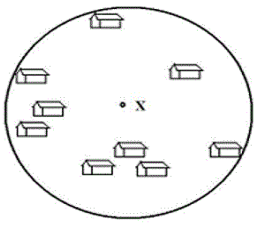

Fig. 5. Minimum enclosing circle [6].

2.4 Wen and Niu Algorithm

The algorithm, proposed in 2010 by Wen and Niu, marks up the fingertips on the contour calculating the angle of each point. The contour of the hand includes only the palm and fingers. The forearm is removed by covering it in swaddling. In order to identify the closest points on a contour, the angle of a point needs to be defined. This can be done by means of the following equation (1) [5].

$$\text{Fingertip angle} < X >= (X - X_1) \bullet (X - X_2)/|X - X_1||X - X_2|, \qquad (1)$$

where:

- X1 and X2 are separated the same distance away by X.
- X1 is the previous point with a certain number of points before X.
- X2 is the next point with a given number of points after X.
- The symbol (•) means that the scalar product.
- The value of the fingertip-Angle <X> is between 0 and 1, as it is the cosine value of X1 X2. Figure 6 illustrates the angle of a point [5].

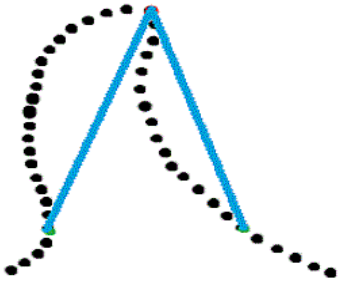

Fig. 6. Angle of a contour point [5].

3 Methodology

COMET (Concurrent Object Modeling and Architectural Design Method), is considered the first methodology for the development of our *LSM* system because it is a methodology that bases its operation on use cases for object modeling and architectural design of concurrent applications, distributed and real-time applications in particular, as is the case of our system.

During the investigation and under the criteria that the COMET methodology provides, it was identified that, during the integration and increment stage of the software, [it?] was not [robust] enough to solve the problem in order to perform the recognition and interpretation of the signs expressed with the hands. In this manner, based on the first tests carried out with the software in Microsoft Visual Studio 2010 with [the] OpenCV tool, which is a free library for artificial vision analysis; it was decided to select another methodology named Automated Object Recognition (*Sistema*

de Reconocimiento Automatizado de Objetos, *SRAO*), complementary to the research by ensuring better results for the system on the part of the video image recognition.

3.1 COMET

It serves as a cycle of software development — based on use cases — working sequentially with the phases that comprise it: development of requirement modeling, analysis modeling, design modeling, and construction and incremental integration of the software in an iterative development cycle, until the validation phase of the system. [10].

3.1.1 Description of the Phases of the COMET Methodology

This methodology consists of the following phases:

- Requirement Modeling: We will have the support of a person who performs *LSM* signs, and a capture device will be used for the representation of the movements made by the user. [10].
- Analysis Modeling: The movements detected by the image capture device will be studied, based on user requirements (initial prototype), in order to then analyze each signal captured with an appropriate pattern recognition algorithm. (Medina, s. f.)
- Design Modeling: Once the image to be processed is captured, it will be analyzed and directed to the knowledge base where the respective sign will be compared. [10].
- Construction and incremental software integration: Through the image capture device, the input image of the human body will be taken, and [then] it will be processed in real time, by means of various algorithms, in order to obtain the skeleton of the image and thus build an approximation based on the results of the image analysis. Once having obtained the latter [results?], they [in turn] will serve to structure the new prototype (Incremental Prototype), this based on the initial characteristics and new results of the algorithm. [10].
- Validation of the system: The system should validate the entry of new data and compare that the results are true, in order to determine the corresponding translation of Mexican Sign Language to written form, returning the text that corresponds to the image on screen. If validation is not carried out successfully, the results are once more assigned to the incremental prototype phase, compared with established features and directed to a new construction in software integration. [10].

3.2 AORS Methodology

The problem that arises with automatic object recognition is to identify and label objects that the capture receives. More specifically, the problem can be described as follows: given an image or capture containing one or more objects of interest, including the background and a set of labels (one for each region of the image), the system must assign labels that correspond to known models or set of regions in the image [11].

3.2.1 Phases of the AORS Methodology

A System for Automatic Object Recognition (AORS) has the following stages:

- **Detection**
- **Acquisition of characteristics**
- **Tracing**
- **Interpretation**

a) **Detection** The purpose of this module is to obtain the image by means of the webcam in real time and retrieve regions that indicate the recognition of a pattern. In this case, our object of interest, a hand. This functionality is accomplished through a learning process focused on a collection of images scanning all along the image until the calculation of the model is obtained. The function is to divide the image into subgroups of pixels. Where each subgroup approximates, in shape and number of pixels, the region of each of the objects in the image. The instant that the detection is satisfactory, that is, that the region of interest has been obtained, then we proceed to the next stage "Getting the characteristics".

b) **Obtaining [of] Characteristics.** This step permits the calculation of the contour of the hand, then the calculation allows to obtain its structural points, and — based on these points — a number of features in common are sought in order to verify that the region actually corresponds to a hand. Another feature that this module includes is [the possibility of] improving the image quality for the processing and calculations to be obtained in the following stages and thus have better results. Operations on the image are performed in order to detect the presence of known objects in the image, these operations are applied to the subsets formed in the segmentation stage. As well as the usage of filters or other methods to reduce noise in images. Feature extractors or operators depend on the type of objects to be recognized and on the models stored for reference.

c) **Tracing.** Once the previous stage "Acquisition of Characteristics" has been executed successfully, one proceeds to the next step which is the tracing of structural points previously obtained along the image [in turn] acquired from the camera.

d) **Interpretation.** Lastly, and after having followed the structural points correctly, the interpretation is carried out, as to the position of structural points. This allows the recognition of gestures and signs of the hand, which are considered as displacements, rotations, finger flexions or the combination of all, to finally take action when they occur. It is worth mentioning that for each incoming image in the camera, the monitoring and interpretation stage will run continuously until a gestural movement is not recognized; in which case the algorithm will restart automatically returning to the execution stage.

4 Digital image Preprocessing Stages

The first step is the change of color space, from an RGB to an HSV color space, in order to make the image capture less sensitive to the brightness of the environment; this is achieved with a function of the OpenCV library that performs the procedure.

Binarization was subsequently implemented with different functions belonging to OpenCV within the HSV color space, as well as previous parameters for image segmentation.

4.1 Extraction of Characteristics

The characteristics extraction problem was solved as follows: first the input is a video capture in real time and the output was established by a set of patterns calculated over the video capture. When the image/video capture is obtained, a treatment is applied to it, with the color space (RGB to HSV), [with a] morphological application (erosion and dilatation) in order to eliminate noise and segmentation (binarization); with the latter we have the discriminated areas (black) and of interest (white); then the Pavlidis algorithm is applied using a function belonging to OpenCV in order to get the coordinates on the border between white and black pixels, and then have them stored in vectors (of points), as a consequence of all this the contours of the object of interest, in this case the hand, is obtained.

With vectors of points in coordinates, [and] in order to detect borders, two more functions are additionally used, both also belonging to OpenCV, with which the convex hull and the enclosing circle will be obtained, the latter two are described below:

1. For the convex hull, a counter-clockwise scan is executed, which takes care of enveloping the previously detected contour, subsequently the center of the object is sought by [means of] the function of the minimum enclosing circle. Due to the implementation of these functions, patterns of distances between the center and the contour of the hand can be obtained, the latter help detect convexity defects. Figure 7.

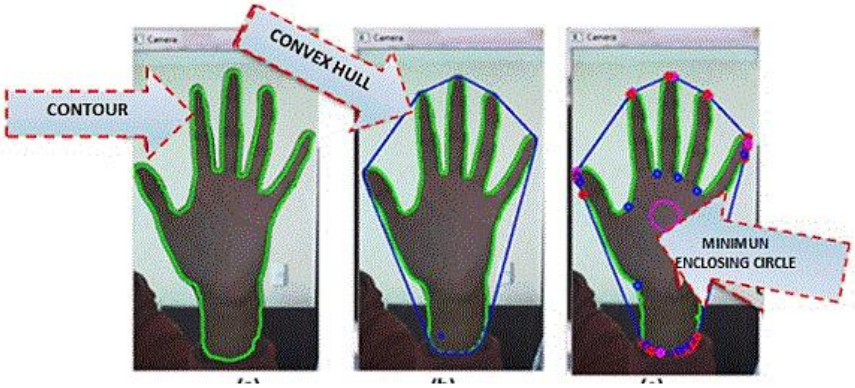

Fig. 7. (a) Contour. **(b)** Convex hull. **(c)** Minimum enclosing circle.

2. The convexity defects were tested as follows: for each defect of convexity there are 4 types of points, the first is the "startpoint", "endpoint", "farpoint ", and finally "depthpoint ", the latter determines the farthest distance between the contour and the convex hull; therefore, with this set of points will obtain specific characteristics, in this case patterns to identify each signal from the *LSM* made by the object of analysis, in this case the hand.
3. To obtain the pattern count (points) a list of these will be made, named F (n), so there will be from F1 to F24, this will be done heuristically, the number of comparisons of variables being 24 (depth, pt.Start.y, ptEnd.y, ptFar.y, ptStart.x, ptEnd.x and ptFar.x); against the constants (mycenter.x, mycenter.y and 11), using for this the operators "greater than" and "less than", so the hand zone will be divided into several areas about its center — as shown in figure 8 —, then the F (n) that are not useful — because they show a large number of patterns (points) — are discarded.

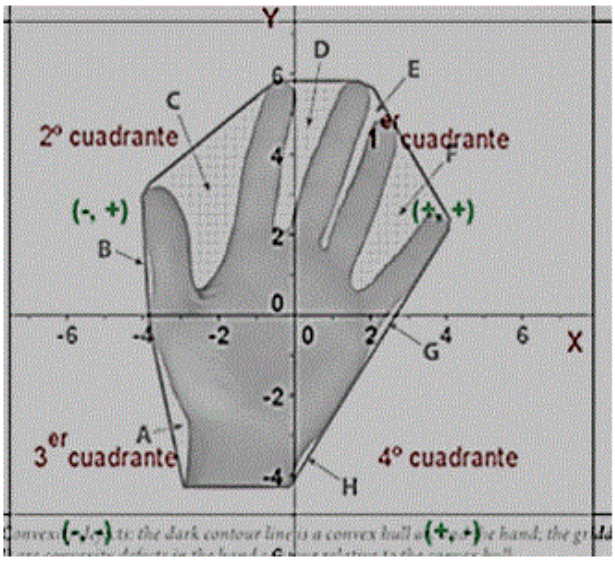

Fig. 8. 1st quadrant, 2nd quadrant, 3rd quadrant, 4th quadrant.

Below, some of the validations of F (n) that were functional for the signs to be detected are shown:

F1: If (depth > 11 && pt.Start.y < mycenter.y) (2)
F2: If (depth > 11 && ptFar.y < mycenter.y) (3)
F3: If (depth > 11 && ptEnd.y < mycenter.y) (4)
F13: If (depth > 11 && pt.Start.x < mycenter.x) (5)
F14: If (depth > 11 && ptFar.x < mycenter.x) (6)
F15: If (depth > 11 && ptEnd.x < mycenter.x) (7)

NÚMERO	# IMAGEN	FINGERS											
		F1	F2	F3	F4	F5	F6	F13	F14	F15	F16	F17	F18
1	Imagen1	1	1	1	1	2	2	1	2	2	2	1	1
	Imagen2	1	1	2	3	4	3	2	2	2	3	3	3
	Imagen3	1	1	2	1	2	1	1	2	2	2	1	1
	Imagen4	0	1	1	2	3	3	1	2	2	3	2	2
	Imagen5	0	1	1	2	2	2	1	2	2	2	1	1
	Imagen6	0	1	1	2	2	2	1	2	2	2	1	1
	Imagen7	0	1	1	2	2	2	1	2	2	2	1	1
	Imagen8	0	1	1	2	2	2	1	2	2	2	1	1

(a)

PALABRA	# IMAGEN	FINGERS											
		F1	F2	F3	F4	F5	F6	F13	F14	F15	F16	F17	F18
ABAJO	Imagen1	1	1	1	1	1	1	1	1	0	1	1	2
	Imagen2	1	2	2	1	1	1	2	2	1	1	1	2
	Imagen3	1	1	1	1	1	1	1	1	0	1	1	2
	Imagen4	1	1	1	1	1	1	1	1	0	1	1	2
	Imagen5	1	1	1	1	1	1	1	1	0	1	1	2

(b)

LETRA	# IMAGEN	FINGERS											
		F1	F2	F3	F4	F5	F6	F13	F14	F15	F16	F17	F18
A	Imagen 1	1	1	1	0	0	0	0	1	1	1	0	0
	Imagen 2	1	1	2	1	1	0	0	1	1	2	1	1
	Imagen 3	1	1	1	0	0	0	1	1	1	0	0	0

(c)

Fig. 9. Matrices of structural values. **(a)** Number Matrix. **(b)** Word Matrix. **(c)** Letter Matrix.

5 Results

The results of the tests to verify that the contours that belong to a hand — by calculating structural points — are based on heuristic conditions where three types of points of convexity defects were considered: the starting point, (which is located where the outline of the hand starts), the convex hull, and the end point (which in this case is the last point of the series before the points once again converge). The *depth* variable is assigned to the end point — which determines the maximum distance between the contour and the convex-hull — to finally assign that point to the place of maximum distance iteratively, until concluding the recognition of the hand. It must be taken into account that different thresholds and logical operators were used. Another point to

consider for the optimizing of the identification of the object is to calculate the minimum enveloping circle, which is used to determine the approximate center of the object of interest (hand). This based on coordinates of the Cartesian 'X', 'Y' Plane and numeric variables, in order to define the threshold conditions, [and in turn] in order to separate the "plane" by quadrants ('y' positive / negative and 'x' positive / negative). It is highly relevant to separate (by quadrants) the convexity points (patterns) by means of heuristics. All of the above based on the depth and the minimum enveloping circle, using the variables: *mycenter*, starting point (*ptStart*) and *depth*:

$$\text{if (depth} > 11 \;\&\&\; \text{ptStart.y} < \text{mycenter.y)}. \tag{8}$$

Where C, D, E and F indicate the depth, the reason for which the threshold is taken as higher than 11 (based on the image pixel count), the other depths being discarded, as to ptStart.y <mycenter. y, [it] will take the points higher than mycenter.y, in this case quadrants 1 and 2, after this other conditions will be used using the above variables, so that the convexity points can be used in all quadrants Furthermore — for better accuracy — we opted for the use of [either] a red or [an] orange glove, as these, in the HSV color space is easier to detect, additionally it can be distinguished from the environment where the video is being captured.

To achieve the recognition of the signs for letter, number and word, a set of values — which represent the structural points of the hand (F1, F2, F3, F4, F5, F6, F13, F14, F15, F16, F17 and F18) — were used, as shown in figure 10.

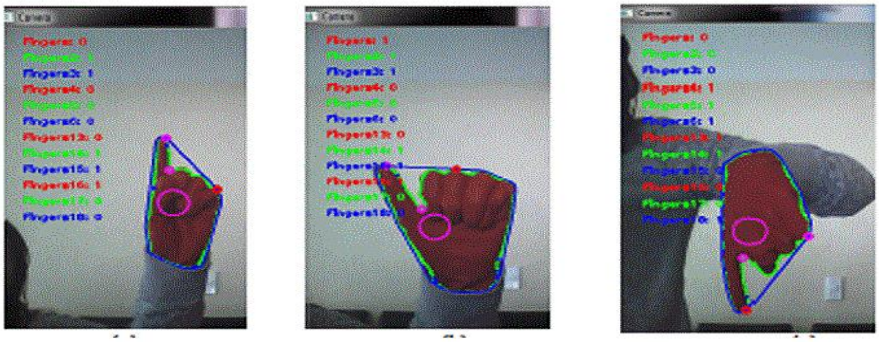

Fig. 10. Structural points. **(a)** Number. **(b)** Letter A. **(c)** Word Below (Abajo).

For the acquisition of the structural points a series of images is captured where the above mentioned, are distinguished (Figure 10)**,** such values are stored in an nX12 matrix, where n is determined when finding a repeatability or constant in the values, thus generating matrices with different numbers of rows, and 12 is the number of structural points used to limit the hand.

It should be mentioned that the functionality of the system will be taken as satisfactory by means of the percentage that its accuracy yields. Recognition is performed based on the counting of the fingers and the results report an accuracy percentage between 50 - 100%; percentages are based on executions that are carried

out in the program, in this case 10 executions which were taken of which if the established sign was right in the 10 it is taken as 100%, if one of those 10 executions only 9 were right 90% taken and so on the accuracy percentage decreases according to the number of times it was right within these 10 runs. The results obtained for a total of 8 numbers, 20 letters of the alphabet and 25 words follow.

Table 1. Percentages of the test of numbers.

NUMBER	PERCENTAGE %
1	90
2	90
3	80
4	90
5	90
6	60
7	70
8	70

Average for the case of the analysis of the numbers

$P_N = \frac{640}{8} = 80\%$

Table 2. Average for the case of the analysis of letters.

LETTERS	PERCENTAGE %
A	90
B	80
C	80
D	70
E	60
F	70
G	60
H	60
I	60
L	70
M	80
N	70
O	70
P	60
R	50
S	60
T	60
U	70
V	70
W	80
Y	60

Average for the case of the analysis of letters:

$P_L = \frac{1430}{21} = 68\%$

Table 3. Average for the case of the analysis of words.

WORDS	PERCENTAGE
ABAJO	70
ADENTRO	60
CEREZA	70
CISNE	50
CONSUEGRO	50
COPA	60
HOGAR	50
JIRAFA	50
MADRE	50
MANO	50
MENOS	60
MESA	60
MUCHO	50
MUÑECA	50
OTOÑO	50
PESOS	50
PISTOLA	70
RESTA	70
SOLO	50
SUMA	50
TORTA	50
UÑA	50
VACA	70
YO	60

Average for the case of the analysis of words:

$$P_L = \frac{1350}{25} = 54\%$$

6 Conclusions

The automatic translation of sign language (*LMS*) is a tool capable of recognizing number, letters and words [expressed] in the *LSM* alphabet, at the end of the development [of the project] the system efficiency was found to have been limited by numerous difficulties that have been encountered during processing of the images or signs captured in real time, such as conducting a training [of the program] for the detection of all the words in sign language, which has forced us to define some parts of the human body that would more easily help identify those signs, but [in turn] would generate a greater degree of complexity in its implementation, [by] adding further convexity points of interest, or otherwise, [by] employing a data base, linked and synchronized to the capture of image data in real-time. Therefore, the use of both hands was restricted, and only the detection with one hand was considered.

As could be seen in the results of tables 1, 2 and 3, these values were very wide-ranging, because the patterns to be identified were becoming confusing by the increase

in the signs to be recognized, for example in the analysis of the numbers, 8 cases were considered where additionally the contour patterns of these signs were well defined, [thus] allowing to obtain a percentage of reliability of approximately 80%; while in the [case of the] letters, in addition to having analyzed more cases (21), the percentage of reliability was lower, this because of several situations, an example is the case of the letter "R", where the middle and index fingers overlap, and the system confuses this pattern as a single point instead of two, because of the crossing of the fingers. In the case of the words "*consuegro*" (≈in law) and "*cisne*" (swan) the use of both hands is necessary, generating noise with the additional hand, for which reason the percentage of identification in those words was also low.

Another point to consider, is the case of the tracking of the frequency of the points, which was lost when performing the bending of a finger, since when straightening the finger there was no continuity of the tracking as in the case of the words "*muñeca*" (wrist/doll), "*suma*" (sum) and "*uña*" (finger/toe nail), to this a solution it was given [by] using convexity defects obtaining a more robust tracking in which point loss is less during the process. So the regions are divided by a convexity hull and a minimum enveloping circle, thus delimiting region by region of interest, from the fingertips to the lower part of the hand.

References

1. Kelly, D., McDonald, J., Markham, C.: A Person Independent System for Recognition of Hand Postures Used in Sign Language. Pattern Recognition Letters, pp. 31, 1359–1368 (2010)
2. Lopez, L. A., Rodriguez, R. M., Zamora, M. G., San Esteban, S.: My hands speak sign language for the deaf. 1st. Editing, editorial Trillas, Mexico, pp. 305 (2010)
3. Serafin, M. E., González, R.: Hands voice. Dictionary of Mexican Sign Language. Free access, National Council to Prevent Discrimination, Mexico, Available at: http://www.conapred.org.mx/documentos_cedoc/DiccioSenas_ManosVoz_ACCSS.pdf (2011)
4. Pavlidis, T.: Algorithms for Graphics and Image Processing. Computer Science Press, Rockville, Maryland (1982)
5. Chen, W. C.: Real-time palm tracking and hand gesture estimation based on fore-arm contour. Doctoral dissertation, PhD thesis, Master dissertation, Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan (2011)
6. Frank, N., Giraldo, M., González, Camargo, E.: Image Processing Algorithms for satellite with Hough Transform. Revista Visión Electrónica Año 5, No. 2, pp 26–41, July-December (2011)