Journal of Software Engineering and Applications, 2015, 8, 419-430
Published Online August 2015 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2015.88041

Hand Gesture Recognition Approach for ASL Language Using Hand Extraction Algorithm

Alhussain Akoum, Nour Al Mawla

Department GRIT, Lebanese University, Beirut, Lebanon
Email: Hussein_akoum@hotmail.com

Received 16 June 2015; accepted 25 August 2015; published 28 August 2015

Abstract

In a general overview, signed language is a technique used for communicational purposes by deaf people. It is a three-dimensional language that relies on visual gestures and moving hand signs that classify letters and words. Gesture recognition has been always a relatively fearful subject that is adherent to the individual on both academic and demonstrative levels. The core objective of this system is to produce a method which can identify detailed humanoid nods and use them to either deliver ones thoughts and feelings, or for device control. This system will stand as an effective replacement for speech, enhancing the individual's ability to express and intermingle in society. In this paper, we will discuss the different steps used to input, recognize and analyze the hand gestures, transforming them to both written words and audible speech. Each step is an independent algorithm that has its unique variables and conditions.

Keywords

Hand Gesture, American Sign Language, Gesture Analysis, Edge Detection, Correlation, Background Modeling

1. Introduction

Gestures are meaningful body movements which are capable of expressing something in a communication, although gesture finds a place to catalogue itself into non-verbal communication, it prominently reaches well to the other end of communication. Gesture is motion of body that contains information [1]. The straightforward purpose of a gesture is to express gen or interrelate with the surroundings. Motionless gestures are those that undertake a precise posted stance. Activity contains a gesture movement that is distinct. Based on the locality of initiation of sign in the body, it can be considered a hand, an arm, a head or a face gesticulation. This paper is exerted on the first type *i.e*. hand gestures. The probable sub-divisions on the hand gestures are static gesture,

How to cite this paper: Akoum, A. and Al Mawla, N. (2015) Hand Gesture Recognition Approach for ASL Language Using Hand Extraction Algorithm. *Journal of Software Engineering and Applications*, **8**, 419-430.
http://dx.doi.org/10.4236/jsea.2015.88041

dynamic gesture, and static and dynamic gesture [2]. Gesture research is termed as a complex research area, as there exists many-to-one mappings from concepts to gestures and gestures to concepts. The major drawback in pursuing research with gestures is that they are ambiguous and incompletely specified [3].

Natural HGR is one of the very active research areas in the Computer Vision field. It provides the easiness to interact with machines without using any extra device and if the users don't have much technical knowledge about the system, they still will be able to use the system with their normal hands. Gestures communicate the meaning of statement said by the human being. They come naturally with the words to help the receiver to understand the communication. It allows individuals to communicate feelings and thoughts with different emotions with words or without words [4].

In our due time, software for sign language recognition is very imperative and is receiving great attention. Such software not only enhances communication between talking people and silent people, but also provides deaf people the ability to interact quickly and professionally with computers and machines using nothing but their hands. American Sign Language (ASL) is a complete system that is considered both simple and complex. It uses 26 different hand signs each indicating a letter. ASL is more than 200 years old. It was the preferable language of 500,000 deaf throughout the United States which rated it as the fourth most-used language.

This language is gaining attractiveness since it supports and enhances communication with an automated system or human located at a distance. Once the user finishes the gesture, the system needs to be capable of identifying it instantly. This is known as "Gesture Recognition". The target of this effort is to construct a system which can classify particular hand gestures and extract the corresponding literatures. This dynamic system is based on the American Sign Language alphabets (**Figure 1**).

2. Overview on the Process

Computers are invariably used by everyone extensively in today's world; one of the major areas of prominence is the human computer interface. Attempts in making a computer understand facial expressions, speech, and human gestures are paving to create a better human computer interaction [5]. Most of the researchers classified gesture recognition system into mainly three steps after acquiring the input image from camera(s) (**Figure 2**), videos or even data glove instrumented device. These steps are: Extraction Method, features estimation and extraction, and classification or recognition as illustrated in figure below [6].

2.1. Similar Systems

The representation captures the hand shape, position of the hand, orientation and movement (if any). The region of interest *i.e*. hand was identified, from where feature vector was to be framed [7]. The feature vector composed for the American Sign Language standard database samples stored consists of. jpg files of existing database along with a few real-time or home-made images. The keypoints derived from the image are placed in an array. All image pixel values that are greater than zero are considered as keypoints and the keypoints array gets generated (**Figure 3**). The match performance based on similarity measures is not made for every point; instead a dimensionality reduction is done. It is taken as the final feature vector. Only retain the keypoints in which the ratio of the vector angles from the nearest to the second nearest neighbor is more [8].

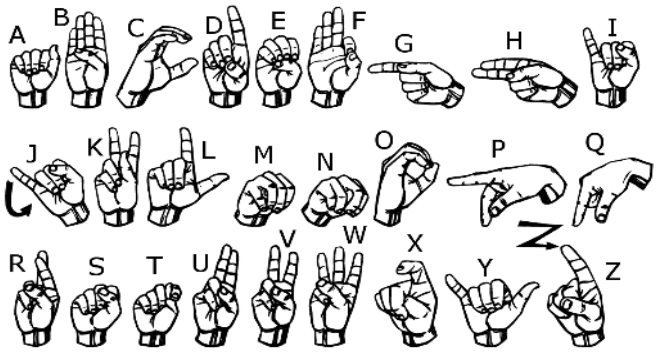

Figure 1. The 26 hand signs of the ASL Language.

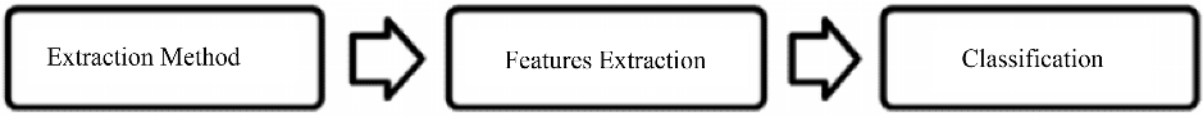

Figure 2. Gesture recognition system steps.

Figure 3. Basic flow of vector composition.

The SIFT detector extracts from an image a collection of frames or keypoints. These are oriented disks attached to blob-alike structures of the image. As the image translates, rotates and scales, the frames track these blobs and thus the deformation. By canonization, *i.e*. by mapping the frames to a reference (a canonical disk), the effect of such deformation on the feature appearance is removed. The SIFT descriptor is a coarse description of the edge found in the frame. Due to canonization, descriptors are invariant to translations, rotations and scaling and are designed to be robust to residual small distortions.

Considering the experimental results of the system above, 80% of the test sample was used for training and 20% for testing. The implementation gave 100% accuracy in identifying the test sample for this dataset only for sample images shown in **Figure 4**. The recognition percentage started to gradually decrease reaching 0% in letters that has almost identical shape such as "M" "N" and "S" (**Figure 4**).

However, the system that we worked on proved to follow a different approach to realize a more precise result. Though our method is more divergent and somehow complicated in terms of code, it is as simple as moving the hand in terms of usage. There are two methods used in building our input data (images), Samples (training) and Live Camera (testing), and our database is wide and variant it covers almost all possible hand positions and skin color (**Figure 5**).

The following flow chart demonstrated in **Figure 6** presents the whole arrangement of the algorithms.

2.2. Creating Database

Image databases pose new and challenging problems to the research community. Over the past 40 years, database technology has matured with the development of relational databases, object-relational databases, and object-oriented databases. The core functionalities of classical databases, however, are tailored toward simple data types and do not extend gracefully to nonstructural information. Digital images have a predominant position among multimedia data types. Unlike video and audio, that is mostly used by the entertainment and news industry [9].

Our database contains about 1200 images composed for the American Sign Language standard samples all in the .bmp format and of dimensions 100 × 100. It is important to mention that having a big database means more accuracy, which is likely to increase the recognition percentage. However, working on software, we must take into consideration the quality of the image as well as the overall size of the program. Since all what we care about in the input image is the hand, which won't exceed 100 × 100 dimensions, we resized the database to fit the description, and thus dramatically decreasing the size of the overall code. Note that later in the code, all images would be changed form "(R. B. G.)" to "Binary", so having the images in. bmp format reduces the image's size while preserving all the information we require from the gesture.

2.3. Camera

By camera we are referring to images captured from the digital camera. It is important to mention that the hardware we are working on, especially the camera, is neither professional nor has any profitable aims. Thus, facing certain obstacles and errors is inescapable. Such errors are not the results of the code but the technical insufficiency. After capturing the image from the camera, the followed steps are:

Feature capture and extraction: Diminish the amount of data by extracting relevant information, which is the hand. In this step it is necessary to detect only the hand and to remove all other features for the image captured

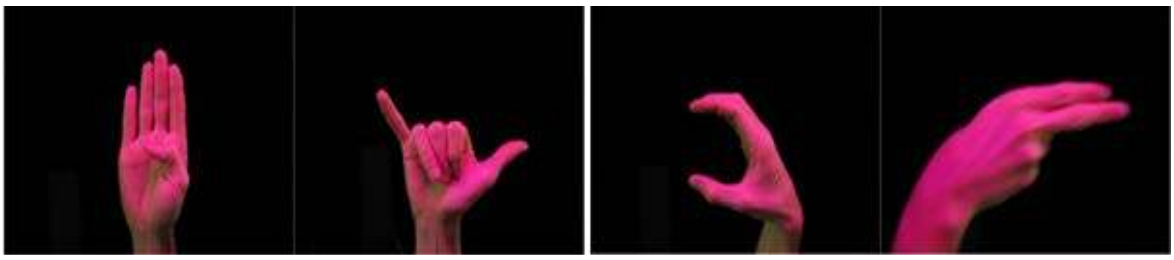

Figure 4. Successful samples.

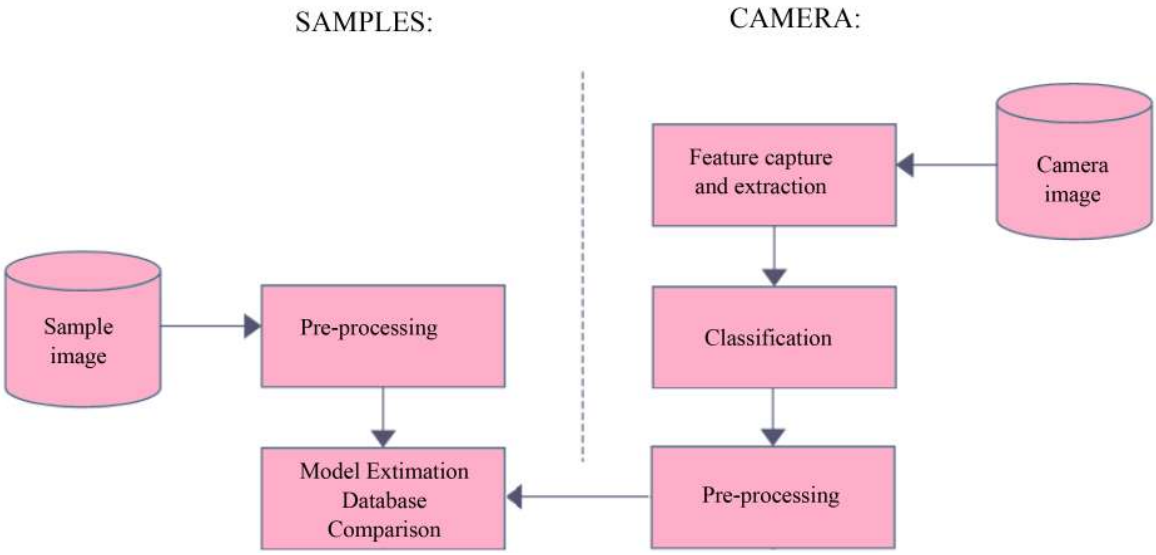

Figure 5. Image classification process.

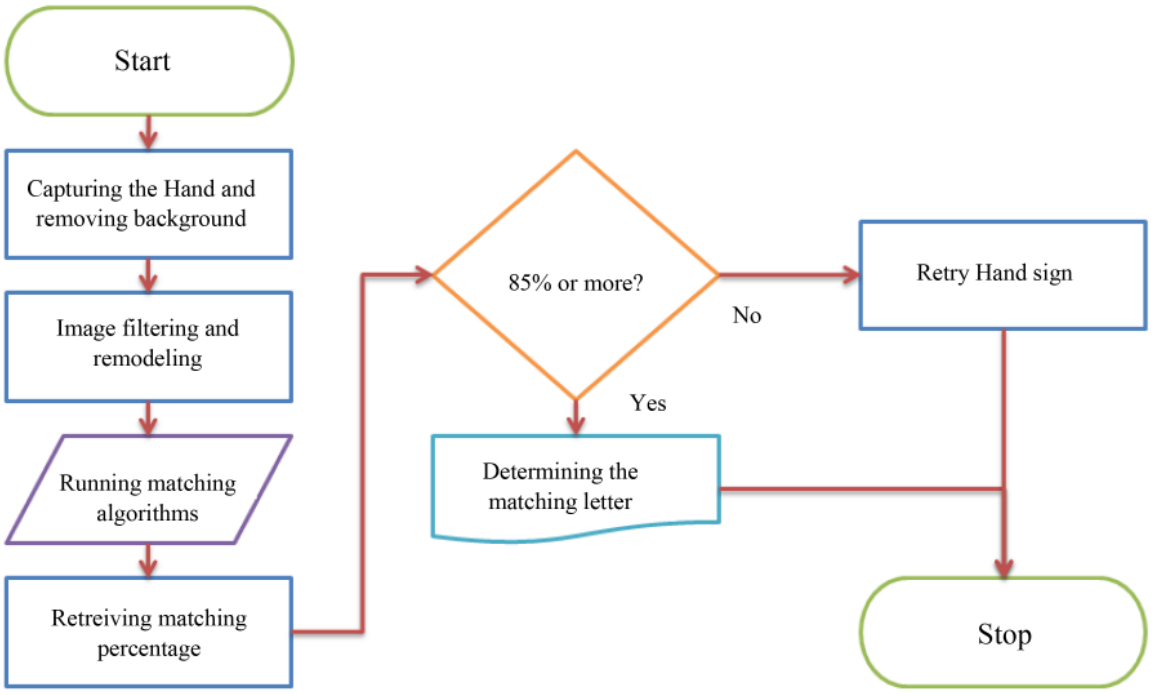

Figure 6. Flow chart of the process.

in the camera. Stabilizing the structures is necessary for distance measurements, *i.e*. the distance between the hand and the camera.

Classification: Before attempting to find a match, the input image must pass through particular functions that change certain properties (color type for example).

Pre-processing: In this step, certain modifications are done on the input image until it is in the appropriate form needed. This step includes trimming, cropping, lining and resizing the image. Given a segmented (isolated) part of the image, features for recognition that should be taken into consideration in a 2-D image are [10]: Total mass (number of pixels in a binarized image), Centroid-Center of mass, Elliptical parameters, Eccentricity (ratio of major to minor axis), Orientation (angle of major axis), Skewness, Kurtosis, Higher order moments, Hough and Chain code transform and Fourier transform and series.

Model Estimation (Database Comparison): Compare the extracted feature image to the various models included in the database and finds the closest match (**Figure 7**).

3. Input Image via Camera

The first step is to capture a plain image of the background before entering the hand gesture. This step is necessary for the upcoming algorithms. The second step is choosing the desired hand gesture. Face the hand straight in front of the camera and make sure your hand is straight and not more than few centimeters away from the camera.

The human steps are done, now it's time for the algorithms to work. Different algorithms and functions are executed after capturing the two required image. The first step is to crop and resize the images.

Removing Background

Background extraction is an important part of moving object detection algorithms that are very useful in surveillance systems. Moving object detection algorithm will be simple by background subtraction when a clean background image is available. The method of extraction the background during training sequence and updating it during the input frame sequence is called background modeling. The main challenges in moving object detection is to extract a clean background and its updating [11].

Thus, removing the background after capturing the image would preserve only the hand (**Figure 8**). The mechanism functions depending on the number of layers, (R. B. G.) Euclidian threshold and fraction of observed color accumulating in the background.

Figure 7. Input camera example.

Figure 8. Input hand before and after background removing.

The above function removed the background but resulted in loss in parts of the hand. This error is mostly due to camera. It is necessary to restore the small area subsequent to the hand itself. To do so, the algorithm changes the image type from (R. B. G.) to "Binary" and applies filtering at different levels. Converting images to binary type is done by replacing all pixels according to the specified luminance with either white (logical 1) if the pixel is equal or greater that the level or black (logical 0) otherwise. Specified level should belong to the range [0, 1]. This level differs from one image to another. Though it can be manually assigned, using the function "Graythresh" to compute it gives better results. The function "Graythresh" function uses Otsu's system in order to determine the threshold of the image and to minimize the interclass modification of the black and white pixels. Multidimensional sections are altered to 2-D arrays The "Graythresh" function overlooks any nonzero invented part of the image. If not specified, the default value of the level is 0.5 (**Figure 9**).

The next step is removing small connected components and objects from binary image. Those objects have fewer pixels than the specified threshold. Since we are working on a 2 dimensional image from a law quality webcam, there is no need to specify any connectivity level. We work with the default connectivity which is 8. Removing the noise in the image is one of the most important and most difficult of the pre-handling techniques. This noise is designated as an unsystematic discrepancy of brightness or color gen through the image's background. 2-D median filtering is one of the most effective functions to overcome many filtering errors. It performs average filtering of the pixels' matrix in both directions. The output pixel comprises the average value of all pixels of the 3-by-3 neighboring region. The procedural steps for 2D median filtering are summarized in the following chart (**Figure 10**).

4. Montage

Perhaps the greatest advantage of our system is that it recognizes not only letters, but also full words. Creating a word using ASL Language means that we need to orderly join multiple images in a single frame. Typically made for this use, we used the montage function, which display all of the structures of a multi-frame image in a particular entity, positioning the frames so that they crudely form a square. What is does is demonstrating a chronological assortment of the specified input images, in other words it changes an array of images into a solo image entity. This function also assembles the borders so that they approximately construct a square. Images can be a series of binary, grayscale, or true-color images. The advantage of this technique is on one hand its ability to specify the word's length, minimizing spelling mistakes, and on other hand preserving all hand gestures and thus reviewing the hand's shape and form throughout the procedure (**Figure 11**).

5. Image Matching

Our task is to recognize a sign based on the input sequence of gestural data, so this is a multiclass classification problem (95 classes). We need to identify the basic characteristics of sign language gestures in order to choose a good classification method. Each sign varies in time and space. Also, the signing speed can differ significantly.

Figure 9. Difference between binary image with (1) Graythresh level (~0.7) and (2) Level = 0.3.

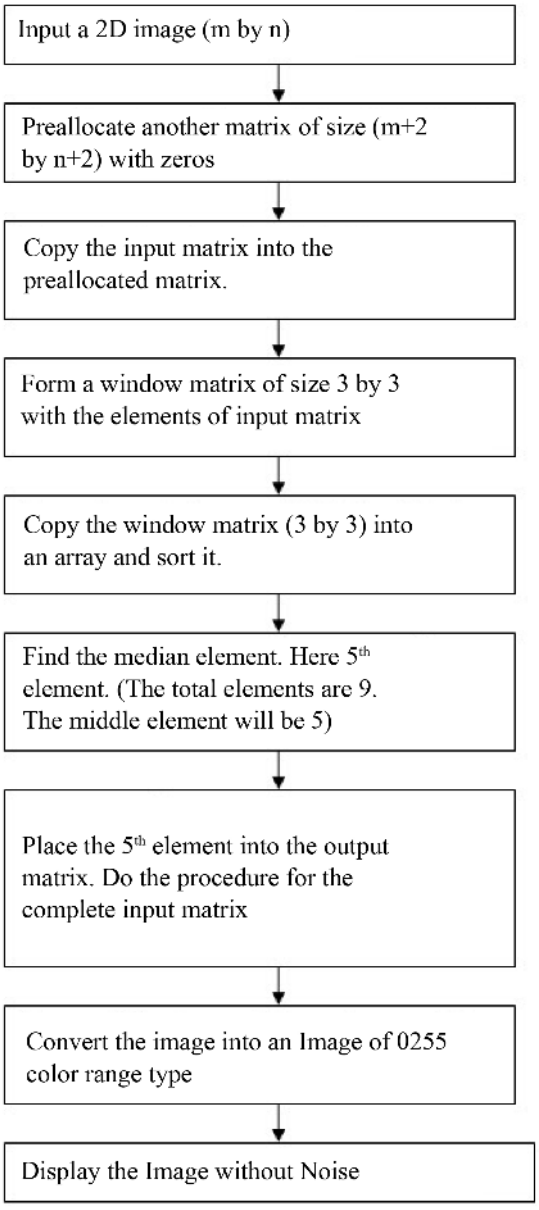

Figure 10. Median filtering.

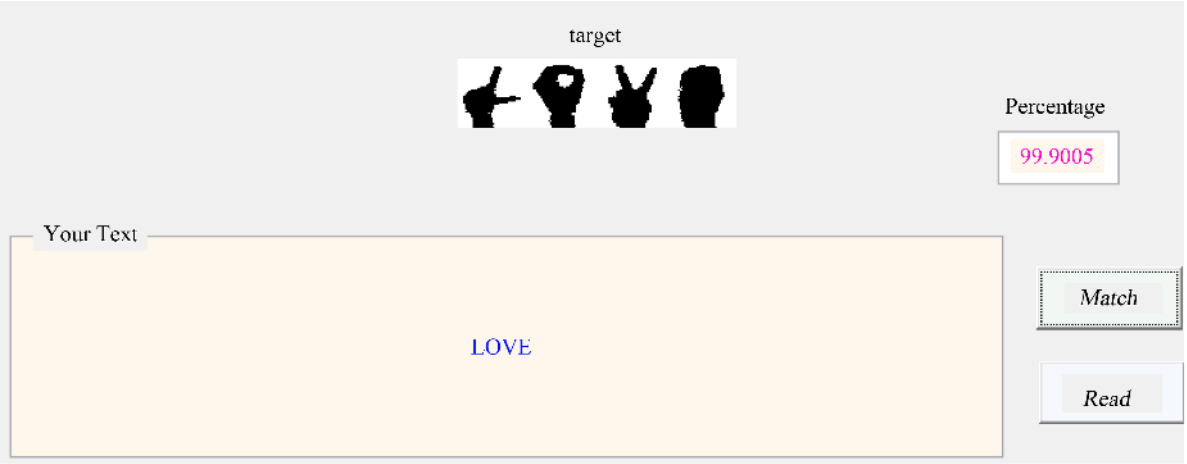

Figure 11. Creating the word "Love".

Even if one person performs the same sign, the speed and position can differ [12].

After creating the database and obtaining the final filtered input image, it's time compare the input image with all images in the database in order to find the closest match. To insure the best results and to receive the closest image after running the matching algorithm, matching is done via 3 different independent methods: 2-D correlation coefficient, Edge detection and Data Histogram.

The matching percentage between input image and each image in the database is stored and eventually the image with highest matching results is chosen.

5.1. Matching via Our Method 1 (2-D Correlation Coefficient)

The correlation coefficient is an integer signifying the resemblance between 2 images regarding their pixel strength. The equation to calculate this coefficient is:

$$r = \frac{\sum_m \sum_n \left(Amn - \bar{A}\right)\left(Bmn - \bar{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(Amn - \bar{A}\right)^2\right)\left(\sum_m \sum_n \left(Bmn - \bar{B}\right)^2\right)}} \quad (1)$$

where $\bar{A} = \text{mean2}(A)$ and $\bar{B} = \text{mean2}$ (2)

A and B are the images meant for matching, while m and n refer to the pixel position in the image Equations (1) and (2). Due to this formula, the size of the two images must be identical, otherwise the result would not be anywhere near accuracy. What the algorithm does is calculate, for every pixel position in both images, the intensity value and compare it to the mean intensity of the whole image. This is type of stabilization method which considers the pixel intensity as the variable to be studied. Eventually, the closer this coefficient is to 1, the closer are the two images. Breaking down an image into blocks before calculating the correlation coefficient reduces the code's elapsing time if the images are relatively large. However, if the correlation coefficient is equal for 2 blocks of images, this doesn't necessarily indicate that images are identical. Using small size images in the database is less likely to cause any error even if resizing a picture may lead to information loss. This method is the one with the most precise results. It provides recognition up to 80% with some errors in extremely close signs. To increase the accuracy, we apply a second level of matching.

5.2. Matching via Our Method 2 Edge Detection

Edge detection is an image handling procedure for finding the borders of entities within images. The mechanism it uses is detecting breaks in the image's illumination. Edge detection is used for image dissection and data abstraction in image processing, computer visualization, and instrumental vision. Applying this method increased matching accuracy from 60% to 80%.

As the name of the method dictates, the matching is based upon finding the image's boundaries. It divides the

image pixel by pixel, find the number of white points (*i.e*. 1 logical) and finally determine the match depending on the average of the number found. The key objective of the edge function is to find the edges intensity of the image. As simple as it may sound, this method is very accurate. Though two images may have same number of white pixels, they are meant to have different edges, and thus are at a certain level of dissimilarity. Using this method to compare one image with a large database of images is quite useful and precise.

Common edge recognition algorithms contain Sobel, Canny, Prewitt, Roberts, and Fuzzy logic methods. Choosing the suitable algorithm for your function is as important as specifying the image's threshold, which identifies its sensitivity (**Figure 12**).

That the system explained above used the Sobel method in the SIFT function. This is an important improvement we provided in our system. It enhanced recognition by 2% - 4%.

Canny edge detection: The explanation of canny procedure is done based on Prof. Thomas Moeslund's research on digital image processing in Indian Institute of Technology [13]. There are five overall steps:

Gaussian filter: As all edge detection consequences are certainly affected by image noise, it is vital to filter out the noise to avoid false recognition (**Figure 13**). This step will faintly level the image to diminish the effects

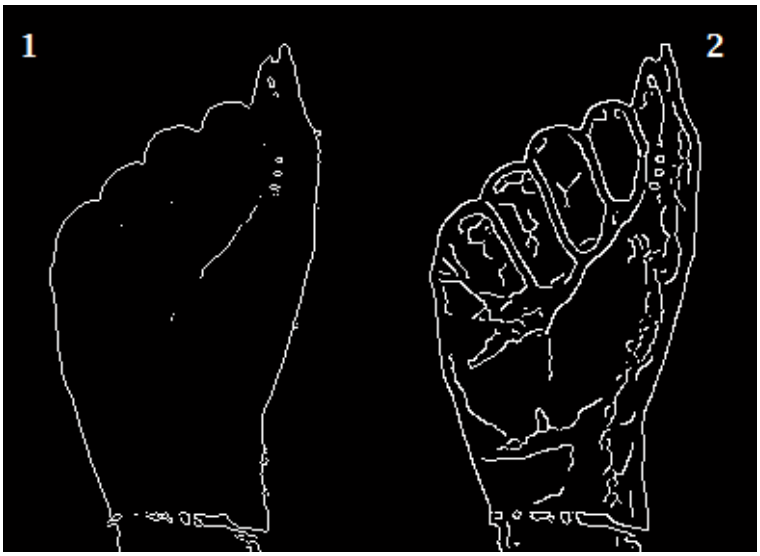

Figure 12. Edge difference between Sobel method (1) and canny method (2).

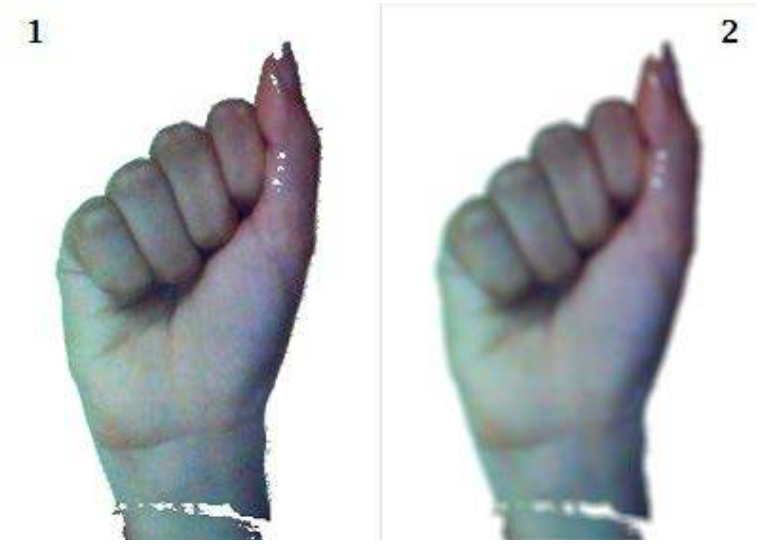

Figure 13. The image before (1) and after (2) a 5 × 5 Gaussian filtering.

of noise during the procedure. The Gaussian filter removes any noise and rough parts of the image.

Intensity Gradient: This step is similar to edging using Sobel method. An edge might point in a range of guidelines, so the Canny algorithm uses four filters to spot horizontal, vertical and diagonal edges in the hazy image resulting from the Gaussian filtering. For each pixel, 2 dimensional convolution matrices are created and in both x and y directions Equation (3).

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Gy = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3}$$

Then, the following equations are applied in order to find the matrix's strength and slope:

$$G = \sqrt{Gx^2 + Gy^2} \tag{4}$$

$$\theta = \arctan \frac{Gy}{Gx} \tag{5}$$

(θ) Represents the direction and the orientation of the edge indicating possible color change. The calculated value subsiding in every color region will be fixed to a specific angle rate, for example θ in yellow region (0˚ to 22.5˚ and 157.5˚ to 180˚) will be set to 0˚ Equations (4) and (5).

Suppression: This step is applied to narrow the edge and eliminate pixels that are not considered to be part of it. Thus, it can help to overturn all the gradient values to 0 excluding the local maximal, which designates position of the piercing change of concentration rate. Compare the strength of the recent pixel and the adjacent pixel in the both directions. If the intensity of the current pixel is the largest compared to the other pixels in the mask with the same direction the value will be conserved. Else, the value will be replaced.

Double threshold: Once suppression is over, the edge pixels are exact to signify the real edge. Nevertheless, at this step there would still be some errors due to color disparity. In order to dispose of them, it is vital to filter out the edge pixel with the minimal gradient value and reserve the edge with the maximal gradient value. Therefore, two threshold values are assigned to elucidate pixels, one is called upper threshold value and the other is called the lower threshold value.

The two threshold values are not simply calculated or assigned. They are specified by the "by trial and error" method. Countless values are applied until the exact one could be determined.

Hysteresis: Hysteresis is the phenomenon in which the values of the physical assets break behind the variations triggering it. To do so, Canny will use the two assigned thresholds (upper and lower) determined in the step above:

- If the pixel's intensity gradient is greater than the upper threshold, the pixel is recognized as an edge.
- If a pixel intensity gradient is less than the lower threshold, then it is excluded.
- If the pixel gradient is between the two thresholds and is adjacent to an edge pixel, it will also be considered to be part of the edge.

This step really makes a big difference in the size of the edge (**Figure 14**).

The figure above clearly shows the dissimilarity in the boundaries' size though the image is binary, which implies that it only has edges when the pixel changes from logical 1 (white) to logical 0 (black).

Finally, and for even more accuracy, a third level of matching is applied.

Matching via method 3 (data Histogram): (Imhist) is an algorithm used to analyze the histogram for the intensity of an image and demonstrates the results in the form of histogram scheme. The quantity of silos in the histogram is itemized by the image form. If the image is grayscale, (imhist) uses a default rate of 256 silos. If it is binary, (imhist) uses two silos.

For concentrated images, each of the n silos of the histogram consists of a half-open interval of width equal to A. $(n-1)^{-1}$ (**Figure 15**).

Finally and after combining the three matching algorithms, the matching results in the unique letters reached 100%, and reached up to 80% in letters that where 0% in other similar systems.

6. Recognizing the Letter

Once running the Matching algorithms, the maximum resemblance percentage between the input image and

Figure 14. Edge difference between Canny (1) and Sobel (2) in a binary image.

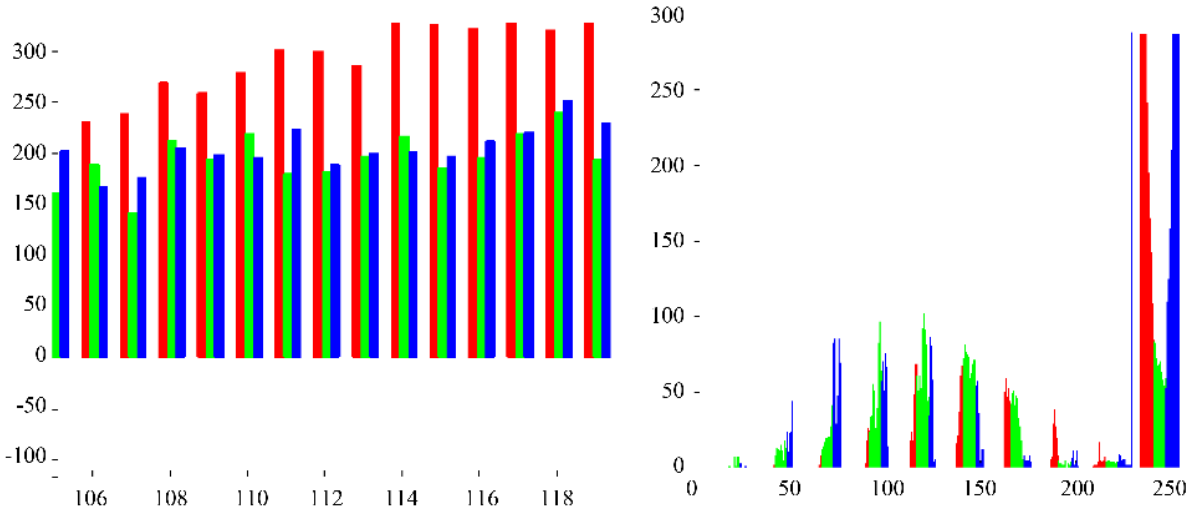

Figure 15. Data histograms of the input image in (RB G) format.

images in the database will be saved and displayed. To realize to which specific image in the database this percentage belongs to, we use the "Find" function which is a built I function used to determine the indices and values of the nonzero elements in the database. After recognizing the index, we need to relate it to a certain letter. For this step, another algorithm is used. For instance, input images that match with the database's image having index "1" will represent the letter "A".

Words containing multiple letters and sentences containing spaces are verified using the same method but after passing through the extraction step which separates each letter and identify empty images as spaces.

7. Text to Speech

A Text-To-Speech (TTS) synthesizer is a computer based system that should be able to read any text aloud, when it is directly introduced in the computer by an operator. It is more suitable to define Text-To-Speech or speech synthesis as an automatic production of speech, by "grapheme to phoneme" transcription. A grapheme is the smallest distinguishing unit in a written language. It does not carry meaning by itself. Graphemes include alphabetic letters, numerical digits, punctuation marks, and the individual symbols of any of the world's writing systems. A phoneme is "the smallest segmental unit of sound employed to form meaningful utterances" [14]. TTS creates speech from a string of characters, and states it. The audio layout is mono, 16 bit, 16 kHz by default. This function requires the Microsoft Win32 Speech API (SAPI).

8. System Results and Advantages

The overall matching result is 85% - 90% recognition for words and letters. It is important to mention that though the system contains up to 1200 images as a database, and about 200 sample images, the system's size doesn't exceed 4 MB. Running software with such a small size is unique and successful. Another advantage is

the speed of the system. Matching a letter with 1200 images using 3 different methods means that the database is processed 1200 × 3 times. However the results are out within few seconds, which means that the system is fast and real-time.

9. Conclusion

Nowadays, hand gesture recognition had been implemented into many forms of bids and in several. This is a proof of the importance and improvement of this research title over the past few years. Hand gestures are at the heart of vision collaboration and machinery control. This paper principally dedicated to the ASL system. Though the equipment supply where is humble and standard, the final results live up to our expectations, we are simply proud of our work. The collection of detailed algorithm for data extracting and matching depends on the request desired. Description of gesture recognition topics and detail debate of used schemes are given as well.

References

[1] Kurtenbach, G. and Hulteen, E.A. (1990) Gestures in Human-Computer Communication. In: Laurel, B., Ed., *The Art of Human-Computer Interface Design*, Addison-Wesley Publishing Company, Inc., New York.

[2] Ibraheem, N.A. and Khan, R.Z. (2012) Vision Based Gesture Recognition Using Neural Networks Approaches: A Review. *International Journal of Human Computer Interaction* (*IJHCI*), **3**, 1-14.

[3] Ferdousi, Z. (2008) Design and Development of a Real-Time Gesture Recognition System. U.M.I. Publishers.

[4] Kendon, A. (2004) Gesture: Visible Action as Utterance. Cambridge University Press, Cambridge.

[5] Yang, M.-H. and Narendra, A.(2001) Face Detection and Gesture Recognition for Human-Computer Interaction. Springer, US. http://dx.doi.org/10.1007/978-1-4615-1423-7

[6] Khan, R.Z. and Ibraheem, N.A. (2012) Hand Gesture Recognition, a Literature Review. *International Journal of Artificial Intelligence & Applications* (*IJAIA*), **3**, 161.

[7] Rokade, U.S., Doye, D. and Kokare, M. (2009) Hand Gesture Recognition Using Object Based Key Fram Selection. *Proceedings of the* 2009 *International Conference on Digital Image Processing*, Bangkok, 7-9 March 2009, 288-291.

[8] Nachamai, M. (2013) Alphabet Recognition of American Sign Language a Hand Gesture Recognition Approach Using Sift Algorithm. *International Journal of Artificial Intelligence & Applications* (*IJAIA*), **4**, 105-115.

[9] Castelli, V. and Bergman, L.D. (2002) Image Databases: Search and Retrieval of Digital Imagery. John Wiley & Sons, Hoboken.

[10] Szmurlo, M. (1995) A Comparative Study of Statistically Classifiable Features Used within the Field of Optical Character Recognition. Master's Thesis, Image Processing Laboratory, Oslo.

[11] Mohamad, H.S. and Mahmood, F. (2008) Real-Time Background Modeling Subtraction Using Two-Layer Codebook Model. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 19 March 2008, 978-988.

[12] Bauer, B. and Hienz, H. (2000) Relevant Features for Video-Based Continuous Sign Language Recognition. *Proceedings of the* 4*th IEEE International Conference on Automatic Face and Gesture Recognition*, Washington DC, 26-30 March 2000, 440. http://dx.doi.org/10.1109/afgr.2000.840672

[13] Moeslund, T. (2009) Canny Edge Detection.

[14] Wright, O. and Wright, W. (2013) Flying-Machine. *International Journal of Advanced Trends in Computer Science and Engineering*, **2**, 269-278.