CHAPTER 2

Feature Extraction Technique for Static Hand Gesture Recognition

Haitham Badi, Sameem Abdul Kareem and Sabah Husien

The goal of static hand gesture recognition is to classify the given hand gesture data represented by some features into some predefined finite number of gesture classes. The main objective of this effort is to explore the utility of two feature extraction methods, namely, hand contour and complex moments to solve the hand gesture recognition problem by identifying the primary advantages and disadvantages of each method. Artificial neural network is built for the purpose of classification by using the back-propagation learning algorithm. The proposed system presents a recognition algorithm to recognize a set of six specific static hand gestures, namely: Open, Close, Cut, Paste, Maximize, and Minimize. The hand gesture image is passed through three stages, namely, pre-processing, feature extraction, and classification. In the pre-processing stage some operations are applied to extract the hand gesture from its background and prepare the hand gesture image for the feature extraction stage. In the first method, the hand contour is used as a feature which treats scaling and translation problems (in some cases). The complex moments algorithm is, however,

Haitham Badi, Sameem Abdul Kareem and Sabah Husien
Faculty of Computer Science & Information Technology,AI Dept.,University of Malaya
50603 Kuala Lumpur, Malaysia
e-mail: haitham@siswa.um.edu.my

Editor: A. Chaudhary, *Recent Trends in Hand Gesture Recognition*

DOI: 10.15579/gcsr.vol3.ch2, GCSR Vol. 3, pp. 19-41, 2015

used to describe the hand gesture and treat the rotation problem in addition to the scaling and translation. The back-propagation learning algorithm is employed in the multi-layer neural network classifier. The results show that hand contour method has a performance of 71.30% recognition, while complex moments has a better performance of 86.90% recognition rate.

2.1 Introduction

Current researches in the field are limited to the use of glove-based, non-skin color background and orientation histogram with their associated problems. In this research we have shown that the use of feature extraction method together with a neural network classifier is able to recognize a limited number of gestures accurately. We have also shown that the recognition system developed is non-costly with respect to time as compared to systems using the Gabor filter [12]. Furthermore, hand recognition system based on the feature extraction method works well under different lighting conditions while most feature extraction methods employed by other researchers [10] failed. We have also shown that the method is employed are able to overcome the limitations of scaling, translation and rotation, associated with most feature extraction methods [1]. This research contributes, in general, to the use of a "natural" mean, namely, hand gesture that humans employ to communicate with each other into Human Computer Interaction (HCI) technology which has become an increasingly important part of our daily lives. Some of the contributions of this research can be summarized as follows:

1. This study develops a static hand gesture recognition system that can be used for different applications which involve a limited number of hand gestures such as virtual mouse.

2. This study investigates the suitability of two different feature extraction approaches to solve the hand gesture recognition problem by identifying the primary advantages and disadvantages of each method. This comparison may shed some lights on the challenges and opportunities that have to be considered if anyone would like to further improve the proposed methods or chooses to employ them in any real-life application. Complex moments method, due to its higher accuracy, is preferred for hand gesture-based desktop applications where the time cost is not of a prime concern while hand contour is better used in hand gesture-based online applications as it is faster in training compared to complex moments.

3. Complex moments method apparently overcomes the challenges other previous methods could not handle, namely, working under different conditions such as scaling, translation and rotation. Hand contour method, however, does not handle rotation cases which limits its use in hand gesture-based applications.

4. Neural network classifier, which is used as a recognizer for hand gesture images based on the features extracted by the two methods, is also evaluated in terms of accuracy, convergence speed and overfitting.

5. The two proposed feature extraction methods can be used by other researchers to develop hand gesture recognition systems or enhance the recognition rate by using new classification methods.

This chapter is divided into four sections. Section 2.2 describes human gestures. This Section also provides the definition of gesture types, and the concept of gesture recognition with its applications. An overview of gesture recognition techniques is also presented, which are used for static and dynamic hand gesture recognition. Section 2.3 presents an overview of the general stages of the system, which includes background information about the image processing, gestures extraction, and neural networks in general. Section 2.4 discusses the experimental results obtained from the presentation of the proposed gesture recognition technique. Section 2.5 highlights the conclusions and provides suggestions for future work.

2.2 Gesture Recognition

2.2.1 Review of Hand Gesture Recognition systems

Gesture recognition is an important topic in computer vision because of its wide range of applications, such as HCI, sign language interpretation, and visual surveillance [20].

Krueger [21] was the first who proposed Gesture recognition as a new form of interaction between human and computer in the mid-seventies. The author designed an interactive environment called computer-controlled responsive environment, a space within which everything the user saw or heard was in response to what he/she did. Rather than sitting down and moving only the user's fingers, he/she interacted with his/her body. In one of his applications, the projection screen becomes the windshield of a vehicle the participant uses to navigate a graphic world. By standing in front of the screen and holding out the user's hands and leaning in the direction in which he/she want to go, the user can fly through a graphic landscape. However, this research cannot be considered strictly as a hand gesture recognition system since the potential user does not only use the hand to interact with the system, but also his/her body and fingers. We choose to cite [21] due to its importance and impact in the field of gesture recognition system for interaction purposes.

Gesture recognition has been adapted for various other research applications from facial gestures to complete bodily human action [7]. Thus, several applications have emerged and created a stronger need for this type of recognition system [7]. In their study, [7] described an approach of vision-based gesture recognition for human-vehicle interaction. The models of hand gestures were built by considering gesture differentiation and human tendency, and human skin colors were used for hand segmentation. A hand tracking mechanism was suggested to locate the hand based on rotation and zooming models. The method of hand-forearm separation was able to improve the quality of hand gesture recognition. The gesture recognition was implemented by template matching of multiple features. The main research was focused on the analysis of interaction modes between human and vehicle under various scenarios such as: calling-up vehicle, stopping the vehicle, and directing vehicle, etc. Some preliminary results were shown in order to demonstrate the possibility of making the vehicle detect

and understand the human's intention and gestures. The limitation of this study was the use of the skin colors method for hand segmentation which may dramatically affect the performance of the recognition system in the presence of skin-colored objects in the background.

Hand gesture recognition studies started as early as 1992 when the first frame grabbers for colored video input became available, which enabled researchers to grab colored images in real time. This study signified the start of the development of gesture recognition because color information improves segmentation and real-time performance is a prerequisite for HCI [32].

Hand gesture analysis can be divided into two main approaches, namely, glove-based analysis, vision-based analysis [16].

The glove-based approach employs sensors (mechanical or optical) attached to a glove that acts as transducer of finger flexion into electrical signals to determine hand posture.

The relative position of the hand is determined by an additional sensor. This sensor is normally a magnetic or an acoustic sensor attached to the glove. Look-up table software toolkits are provided with the glove for some data-glove applications for hand posture recognition. This approach was applied by [27] to recognize the ASL signs. The recognition rate was 75%. The limitation of this approach is that the user is required to wear a cumbersome device, and generally carry a load of cables that connect the device to a computer [28]. Another hand gesture recognition system was proposed in [33] to recognize the numbers from 0 to 10 where each number was represented by a specific hand gesture. This system has three main steps, namely, image capture, threshold application, and number recognition. It achieved a recognition rate of 89% but it has some limitations as it functioned only under a number of assumptions, such as wearing of colored hand gloves and using a black background.

The second approach, vision based analysis, is based on how humans perceive information about their surroundings [16]. In this approach, several feature extraction techniques have been used to extract the features of the gesture images. These techniques include Orientation Histogram [10, 34], Wavelet Transform [35], Fourier Coefficients of Shape [22], Zernike Moment [6], Gabor filter [2, 14, 12], Vector Quantization [24], Edge Codes [13], Hu Moment [39], Geometric feature [5] and Finger-Earth Mover's Distance (FEMD) [31].

Most of these feature extraction methods have some limitations. In orientation histogram for example, which was developed by [23], the algorithm employs the histogram of local orientation. This simple method works well if examples of the same gesture map to similar orientation histograms, and different gestures map to substantially different histograms [10]. Although this method is simple and offers robustness to scene illumination changes, its problem is that the same gestures might have different orientation histograms and different gestures could have similar orientation histograms which affects its effectiveness [19]. This method was used by [10] to extract the features of 10 different hand gestures and used the nearest neighbour for gesture recognition. The same feature extraction method was applied in another study [34] for the problem of recognizing a subset of American Sign Language (ASL). In the classification phase, the author used a Single Layer Perceptron to recognize the gesture images. Using the same feature method, namely, orientation histogram, [16] proposed a gesture recogni-

tion method using both static signatures and an original dynamic signature. The static signature uses the local orientation histograms in order to classify the hand gestures. Despite the limitations of orientation histogram, the system is fast due to the ease of the computing orientation histograms, which works in real time on a workstation and is also relatively robust to illumination changes. However, it suffers from the same fate associated with different gestures having the same histograms and the same gestures having different histograms as discussed earlier.

In [2], the authors used Gabor filter with PCA to extract the features and then fuzzy-c-means to perform the recognition of the 26 gestures of the ASL alphabets. Although the system achieved a fairly good recognition accuracy 93.32%, it was criticized for being computationally costly which may limit its deployment in real-world applications [12].

Another method extracted the features from color images as in [37] where they presented a real-time static isolated gesture recognition application using a hidden Markov model approach. The features of this application were extracted from gesture silhouettes. Nine different hand poses with various degrees of rotation were considered. This simple and effective system used colored images of the hands. The recognition phase was performed in real-time using a camera video. The recognition system can process 23 frames per second on a Quad Core Intel Processor. This work presents a fast and easy-to-implement solution to the static one hand-gesture recognition problem. The proposed system achieved 96.2% recognition rate. However, the authors postulated that the presence of skin-colored objects in the background may dramatically affect the performance of the system because the system relied on a skin-based segmentation method. Thus, one of the main weaknesses of gesture recognition from color images is the low reliability of the segmentation process, if the background has color properties similar to the skin [26].

The feature extraction step is usually followed by the classification method, which uses the extracted feature vector to classify the gesture image into its respective class. Among the classification methods employed are: Nearest Neighbour [10, 22, 6], Artificial Neural Networks [34, 17, 27], Support Vector Machines (SVMs) [14, 39, 12], Hidden Markov Models (HMMs) [37].

As an example of classification methods, Nearest Neighbour classifier is used as hand recognition method in [22] combined with Modified Fourier Descriptors (MFD) to extract features of the hand shape. The system involved two phases, namely, training and testing. The user in the training phase showed the system using one or more examples of hand gestures. The system stored the carrier coefficients of the hand shape, and in the running phase, the computer compared the current hand shape with each of the stored shapes through the coefficients. The best matched gesture was selected by the nearest neighbour method using the MED distance metric. An interactive method was also employed to increase the efficiency of the system by providing feedback from the user during the recognition phase, which allowed the system to adjust its parameters in order to improve accuracy. This strategy successfully increased the recognition rate from 86% to 95%.

Nearest neighbour classifier was criticised for being weak in generalization and also for being sensitive to noisy data and the selection of distance measure [3].

To sum up the related works, we can say that hand gesture recognition systems are

generally divided into two main approaches, namely, glove-based analysis and vision-based analysis. The first approach, which uses a special glove in order to interact with the system, and was criticized because the user is required to wear a cumbersome device with cables that connect the device to the computer. In the second approach, namely, the vision-based approach, several methods have been employed to extract the features from the gesture images. Some of these methods were criticized because of their poor performance in some circumstances. For example, orientation histogram's performance is badly affected when different gestures have similar orientation histograms. Other methods such Gabor filter with PCA suffer from the high computational cost which may limit their use in real-life applications. In addition, the efficiency of some methods that use skin-based segmentation is dramatically affected in the presence of skin-colored objects in the background.

Furthermore, hand gesture recognition systems that use feature extraction methods suffer from working under different lighting conditions as well as the scaling, translation, and rotation problems.

2.2.2 Summary

Hand gestures which are performed by one or two hands can be categorized according to their applications into different categories including conversational, controlling, manipulative and communicative gestures. Generally, hand gesture recognition aims to identify specific human gestures and use them to convey information. The process of hand gesture recognition composes mainly of four stages: (1) hand gesture images collection, (2) gesture image preprocessing using some techniques including edge-detection, filtering and normalization, (3) capture the main characteristics of the gesture images using feature extraction algorithms, and (4) the evaluation (or classification) stage where the image is classified to its corresponding gesture class. There are many methods that have been used in the classification stage of hand gesture recognition such as Artificial Neural Networks, Template Matching, Hidden Markov Models and Dynamic Time Warping.

2.3 Basic Concepts of Image Processing and Neural Networks

The study of gesture recognition and gesture-based interaction is increasingly becoming an attractive research subject in HCI. The implementation of a gesture-based HCI requires the capturing of necessary information to determine what gesture is being performed by the user. Recognizing gestures is a complex task that primarily involves two phases: first, extracting features, which characterises the gestures from the images, and second, using a suitable classifier (in this study, neural network) to assign each gesture to its respective class based on the extracted features. These two phases involve numerous techniques and methods that fall under these two following areas:

- Image processing
- Neural network

The purpose of this section is to introduce some basic concepts and methods related to these two areas that are employed in our methodology.

2.3.1 Segmentation

Segmentation is the initial stage for any recognition process in which the acquired image is broken down into meaningful regions or segments. The segmentation process is only concerned with partitioning the image and not with what the regions represent. In the simplest case (binary images), only two regions exist: a foreground (object) region and a background region. In gray level images, several types of region or classes may exist within the image. For example, when a natural scene is segmented, regions of clouds, ground, buildings, and trees may exist [4]. Segmentation subdivides an image into its constituent parts, the level of which depends on the problem being solved. Segmentation should be stopped when the objects of interest in an application have been isolated [11]. The two main approaches to segmentation are as follows:

1. Pixel-based or local methods, which include edge detection and boundary detection
2. Region-based or global approaches, which include region merging and splitting, and thresholding [4].

2.3.2 Thresholding

A simple image segmentation problem occurs when an image contains an object that has homogeneous intensity and a background with different intensity levels [30]. This problem can be overcome by employing thresholding techniques, such as partitioning the image histogram using a single threshold T. Segmentation is then accomplished by scanning the image pixel by pixel and labelling each pixel as an object or a background depending on whether the gray level of that pixel is greater or less than the value of the threshold T [11].

$$g(x,y) = \begin{cases} 1, & if\ g(x,y) > T \\ 0, & otherwise \end{cases} \tag{2.1}$$

2.3.3 Noise Reduction

Spatial filters can be effectively used to remove various types of noise in digital images. These spatial filters typically operate on small neighbourhoods ranging from (3×3) to (11×11). Numerous spatial filters are implemented with convolution masks, because a convolution mask operation provides a result that is a weighted sum of the values of a pixel and its neighbours. This result is called a linear filter. The mean filters are essentially averaging filters; they operate on local groups of pixels called neighbourhoods and replace the centre pixel with the average of the pixels in this neighbourhood. This replacement is performed with a convolution mask [36]. The median filter is a nonlinear filter. A nonlinear filter gives a result that cannot be obtained by the weighted sum of the neighbourhood pixels as was performed with the convolution masks [36].

However, the median filter does operate on a local neighbourhood after the size of the local neighbourhood is defined. The centre pixel is replaced by the median or the centre value present among its neighbours, rather than by the average [36]. The median filter disregards extreme values (high or low) and does not allow such values to influence the selection of a pixel value that is truly representative of the neighbourhood. Therefore, the median filter is excellent in removing isolated extreme noise pixels (often known as "salt and pepper" noise) while substantially retaining spatial detail. However, its performance deteriorates when the number of noise pixels is more than half the number of pixels in the window [4].

2.3.4 Edge Detection

Edges are basic image features that carry useful information regarding the object boundaries. This information can be used for image analysis, object identification, and image filtering applications [30]. Edge detection methods are used as the first step in the line detection process. Edge detection methods are also used in finding complex object boundaries by marking the potential edge points that correspond to the places in an image where changes in brightness occur. After these edge points are marked, they are merged to form lines and object outlines. Edge detection operations are based on the idea that the edge information in an image can be found by examining the relationship between a pixel and its neighbours. If a pixel's gray level value is similar to those around it, then this pixel is probably not an edge point. By contrast, if a pixel has neighbours with widely varying gray levels, then this pixel may represent an edge point. Thus, an edge is defined by a discontinuity in gray level values. Ideally, an edge is caused by changes in colour or texture or by the specific lighting conditions present during the image acquisition process [36].

Sobel Operator The Sobel operator is recognized as one of the best "simple" edge operators that utilizes two (3×3) masks [4]. The Sobel edge detection masks search for the horizontal and vertical directions and then combine this information into a single metric. The masks are given as follows: Each mask is convolved with the image. Two numbers exist at each pixel location, namely, P_1 and P_2, which correspond to the row and the column masks, respectively. These numbers are used to compute two metrics, namely, the edge magnitude and the direction, which are defined as follows [36]:

$$Edge\,Magnitude = \sqrt{P_1^2 + P_2^2} \tag{2.2}$$

$$Edge\,Direction = \tan^{-1}\left(\frac{P_1}{P_2}\right) \tag{2.3}$$

2.3.5 Coordinate Normalization

The idea of coordinate normalization is to map the scaled hand image coordinates to the standard size ranging between $[-1, +1]$ [25]. The purpose of this step is to keep the domain of the image coordinates fixed irrelevant to the original size. The condition of keeping the domain of the coordinates within the limited boundaries will

effectively satisfy the convergence of higher ordered moments. Thus, the scaled image coordinates (X, Y) will be transformed into the normalized set (X_n, Y_n) which can be considered as the "standard" version of the original coordinate (X, Y). By using the center of the image, each pixel coordinates (X, Y) values are mapped to the domain $[-1, +1]$ which can be performed using the following equations:

$$X_n = (2/(W-1) * X) \tag{2.4}$$

$$Y_n = (2/(H-1) * Y) \tag{2.5}$$

Where H and W are the height and width of the scaled image respectively [25].

2.3.6 Feature Extraction

Feature extraction is part of the data reduction process and is followed by feature analysis. One of the important aspects of feature analysis is determining exactly which features are important [36]. Feature extraction is a complex problem in which the whole image or the transformed image is often taken as the input. The goal of feature extraction is to find the most discriminating information in the recorded images. Feature extraction operates on two-dimensional image arrays but produces a list of descriptions or a "feature vector" [4, 15].

Mathematically, a feature is an n-dimensional vector with its components computed by some image analysis. The most commonly used visual cues are colour, texture, shape, spatial information, and motion in video. For example, colour may represent the colour information in an image, such as colour histogram, colour binary sets, or colour coherent vectors. The n components of a feature may be derived from one visual cue or from composite cues, such as the combination of colour and texture [15].

The selection of good features is crucial to gesture recognition because hand gestures are rich in shape variation, motion, and textures. Although hand postures can be recognized by extracting some geometric features such as fingertips, finger directions, and hand contours, these features are not always available and reliable because of self-occlusion and lighting conditions. Moreover, although a number of other non-geometric features are available, such as colour, silhouette, and textures, these features are inadequate for recognition. Explicitly specifying features is not easy. Therefore, whole images or transformed images are taken as input, and features are selected implicitly and automatically by the classifier [38].

2.3.6.1 Complex Moments (CMs)

The notation of CM was introduced by [1] as a simple and straightforward method of deriving moment invariants. The CM of order (m) is defined as [1]:

$$C_m = \int\int (x+iy)^m \mu(x,y)\, dxdy \tag{2.6}$$

Where $i = \sqrt{(-1)}$ and $\mu(x, y)$ is the real image intensity function. A moment invariant is a set of moment values extracted from the image data in such a way that

their values are invariant to the rotation of the image data. Moreover, the value of a CM could be considered a moment invariant if that value can be computed from a group of CMs for the same object at different resolutions [25]. Thus, a moment invariant can be used as a feature for the classification and recognition of an object. In turn, CMs, which are simple and relatively powerful in providing the analytic characteristics of moment invariants, have been proposed as a solution to different pattern recognition problems. CMs have two parts, namely, real and imaginary parts. However, the computation of their values decomposes into two directions: the x-axis moment, which represents the direction of the real part, and the y-axis moment for the direction of the imaginary part [25]. Moment sets can offer a powerful description of the geometrical distribution of the material within any region of interest. The low order of CMs has meanings that are significantly relevant to a number of well-known physical quantities [25].

1. Zero-order moments represent the total mass of the image.

2. First-order moments together with zero-order moments assign the center of the mass of the image. Second-order moments represent moment of inertia.

3. Third-order and fourth-order moments are used for computing statistical quantities such as skews and kurtosis, respectively.

Although higher nth-order moments provide additional statistical and structural information about an image, these moments are computationally more expensive. The computation of a CM should involve the calculation of its real and imaginary components. The nth-order CM (M_i). for the hand image of size $(n \times m)$ is then calculated according to the following equation [25]:

$$M_i = \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} Z_n^i a\,(x, y) \tag{2.7}$$

where (i) indicates the order of the moment, $Z_n = X_n + iX_n$is a complex number, $a\,(x, y)$ represents a pixel value at the position (x, y), which may be an ON (i.e., its value is 1) or OFF (its value is 0) pixel.

The calculation of complex moments may require a long computation time. The following relation is used to reduce the computation time [25]:

$$Z^n = Z^{n-1}.Z \tag{2.8}$$

where $Z = x + iy$ is the complex number.

When the real and imaginary parts of (Z^n and Z^{n-1}) are assumed as complex numbers, they could be written as [25]:

$$Z^n = R_n + iI_n \tag{2.9}$$

and

$$Z^{n-1} = R_{n-1} + iI_{n-1} \tag{2.10}$$

By considering the case $Z^0$and Z^1, we can obtain

$$\begin{aligned} R_0 &= 1; I_0 = 0 \\ R_1 &= x; I_1 = y \end{aligned}$$

The values of Z^n, Z^{n-1} and $Z^1$are then substituted in Eq.(2.11):

$$R_1 = R_{n-1}X - I_{n-1}Y \tag{2.11}$$

$$I_1 = I_{n-1}X - R_{n-1}Y \tag{2.12}$$

These equations indicate that the knowing components (Z^{n-1}) will be directly used to compute the Z^n component [25].

2.3.7 Artificial Neural Networks

An artificial neural network is an information processing system that has certain performance characteristics similar to biological neural networks. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology on the basis of the following assumptions [8]:

1. Information processing occurs at many simple elements called neurons;

2. Signals are passed between neurons over connection links;

3. Each connection link has an associated weight, which multiplies the signal transmitted in a typical neural network;

4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) [8, 29].

The arrangement of neurons into layers and the connection patterns within and between layers is called the "network architecture." Neural networks are often classified as either single layer or multilayer [29].

2.3.7.1 Single-Layer Neural Networks

A single-layer network has one layer of connection weights. The units can often be distinguished as input units, which receive signals from the outside world, and as output units, from which the response of the network can be read. In a typical single-layer network , the input units are fully connected to the output units, but not to other input units. Similarly, the output units are not connected to other output units [8].

2.3.7.2 Multilayer Network

A multilayer network is composed of one or more layers (or levels) for nodes (the so-called hidden units) between the input and output units. Typically, a layer of weights exists between two adjacent levels of units (input, hidden, and output). Multilayer networks can solve more complicated problems than a single-layer network could. However, training the former may be more difficult than training the latter. Multilayer perceptron neural networks are useful for classification purposes[8, 29].

2.3.7.3 Summary

Hand gesture recognition process involves several techniques and algorithms that fall under the areas of image processing and artificial neural networks. The first phase deals with problems related to image processing, such as reducing noise by using filters, scaling, and break down the image into meaningful regions using segmentation techniques such as thresholding and edge detection methods. A powerful classification method called Artificial Neural Networks is selected to classify the images into their respective classes using the extracted feature vectors. This classification method is inspired by the characteristics of biological neural networks. ANN is generally divided based on the learning paradigm into two categories: supervised and unsupervised neural networks. The multi-layer perceptron which uses back-propagation learning method is one of the most used supervised neural networks [17].

2.4 Experimental Setups

In this section, we will briefly describe and illustrate the steps of the experimental work and then define the values of the parameters for both methods, such as those used for ANN implementation. In addition, a description of the database used for training and testing is presented.

2.4.1 Hand Gesture Image Database

This work started with the creation of a database with all hand gesture images to be used for training and testing. The construction of a database for hand gesture (i.e., the selection of specific hand gestures) generally depends on the intended application. A vocabulary of six static hand gestures is made for HCI as shown in Fig.(2.1).

Each gesture represents a gesture command mode. These commands are commonly used to communicate and can thus be used in various applications such as a virtual mouse that can perform six tasks (Open, Close, Cut, Paste, Maximize, and Minimize) for a given application. The gesture images have different sizes. The images are captured using a digital camera and are taken from one subject. The database consists of 30 images for the training set (five samples for each gesture) and 56 images for testing with scaling, translation, and rotation effects. Employing relatively few training images facilitates the measurement of the robustness of the proposed methods, given that the use of algorithms that require relatively modest resources either in terms of training data or computational resources is desirable [9, 18].

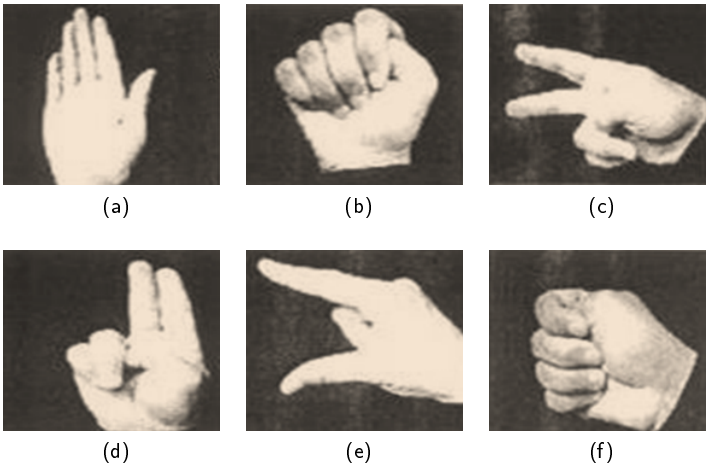

Figure 2.1: Six static hand gestures: (a) Open, (b) Close, (c) Cut, (d) Paste, (e) Maximize and (f) Minimize.

2.4.2 Hand Contour with ANNs

In this stage, many processes were performed on the hand gesture image to prepare these images for the subsequent feature extraction stage. These processes were performed using some image processing operations mentioned in Section 2.3. The effect of these operations is explained below.

2.4.2.1 Hand Gesture Segmentation

All the techniques used in this research are based on hand shapes. The color images of hand gestures were segmented to isolate the foreground (hand) from the background. The results of hand segmentation are shown in Fig.(2.2).

2.4.2.2 Noise Reduction

The segmented images may contain some noises that will affect the results of the feature extraction stage. Thus, a median filter was used to reduce the noise as much as possible. The effect of this filter is shown in Fig.(2.3).

The Edge detection process was performed by using a Sobel operator, as shown in Fig.(2.4).

The result of the pre-processing phase was fed to the feature extraction stage to compute for the feature information of the gesture image. As soon as the contour feature is computed, the image size is adjusted, such that each hand gesture image has the size of 32×32. Image resizing accelerates the system and reduces the negative effects of size change by creating a standard size for all images. An example of the effect of image resizing. The general features (height offset and width offset) will be computed implicitly.

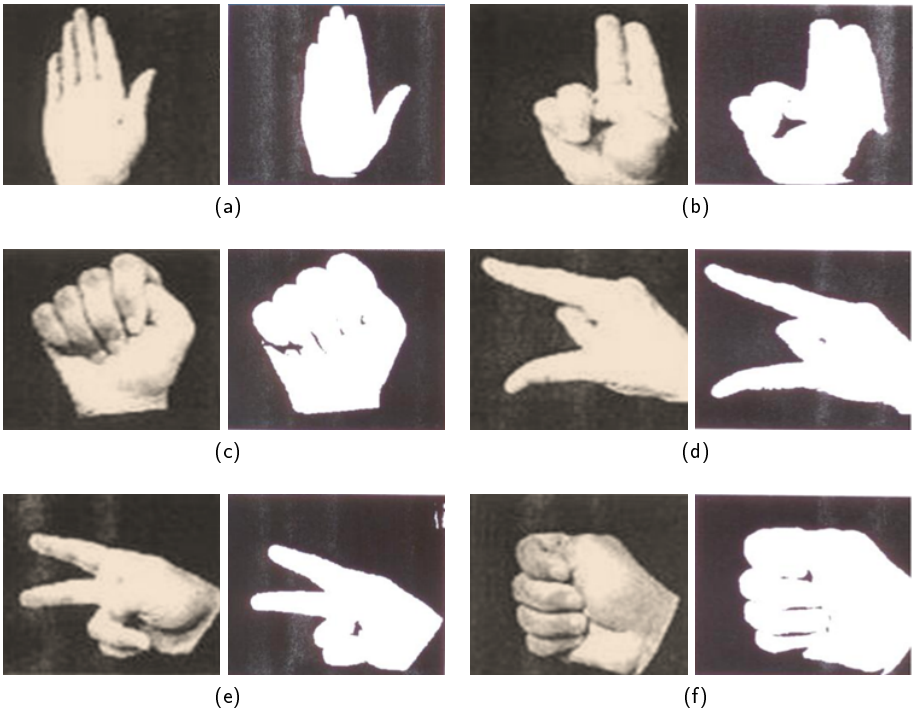

Figure 2.2: Original and the corresponding threshold segmented images of the six static hand gestures: (a) Open, (b) Close, (c) Cut, (d) Paste, (e) Maximize and (f) Minimize.

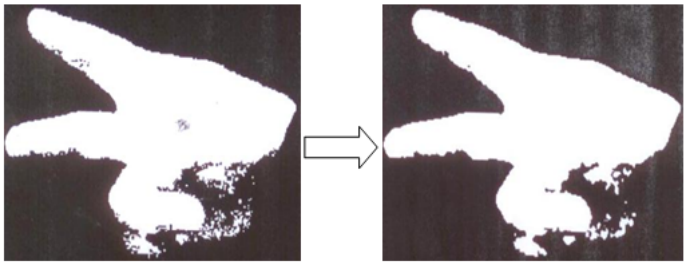

Figure 2.3: Median filter effect.

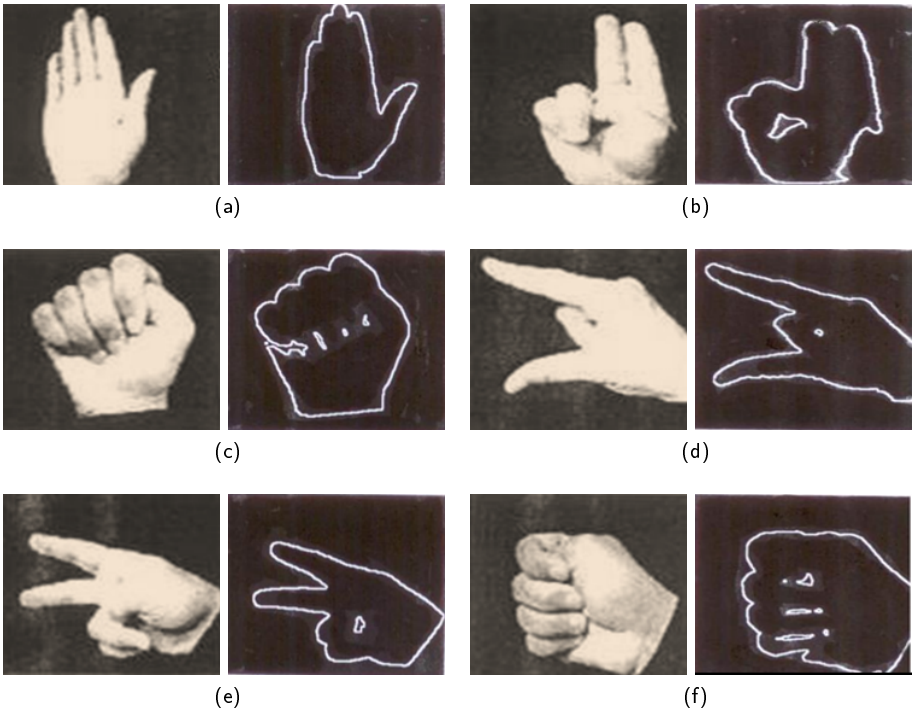

Figure 2.4: Original and the corresponding edge images of the six static hand gestures: (a) Open, (b) Close, (c) Cut, (d) Paste, (e) Maximize and (f) Minimize.

Table 2.1: Parameters for the five Neural Networks.

Parameters	Values
Input Layer	1060 nodes
Hidden Layer	100 nodes
Output layer	6
Stop error	0.01
Learning rate	0.9

2.4.2.3 Training Phase

In this phase, the composite feature vectors computed earlier and stored in a feature image database are used as inputs to train the neural networks in the next stage, as shown in Fig.(2.5). The learning process for the five multilayer neural networks is accomplished by using the parameters shown in Table 2.1.

2.4.2.4 Testing Phase

After training the five neural networks, the performance is evaluated by applying the testing set on the network inputs and then computing the classification error. The activation function used is the binary-sigmoid function, which always produces outputs between 0 and 1. In our case, the five neural networks are used in a sequential manner, i.e., the test gesture feature image will be entered to the first neural network; if the network successfully recognizes the gesture, the test operation stops. If this network does not recognize the gesture features, the second network will be activated, and so on. If all the five networks fail to identify the feature, a message "gesture not recognized" appears. Notably, the failure of the neural network to recognize the gesture rather than wrongly recognizing it is directly related to the output of the network, where the recognized image is the one that receives the highest value; in the case where two or more images receive the same highest output value, the network fails to recognize the gesture.

In the testing phase, 56 hand gesture images were used to test the system under different light conditions and with the effects of scaling and translation. The system is capable of recognizing and classifying any unknown gesture if such gesture is in the original database.

2.4.3 Complex Moment with ANNs

The processing stage in this method includes, in addition to segmentation and noise reduction processes as in the previous method, image trimming for eliminating the empty space and extracting only the region of interest, followed by the normalization process. The effect of these operations will be presented in the next sections.

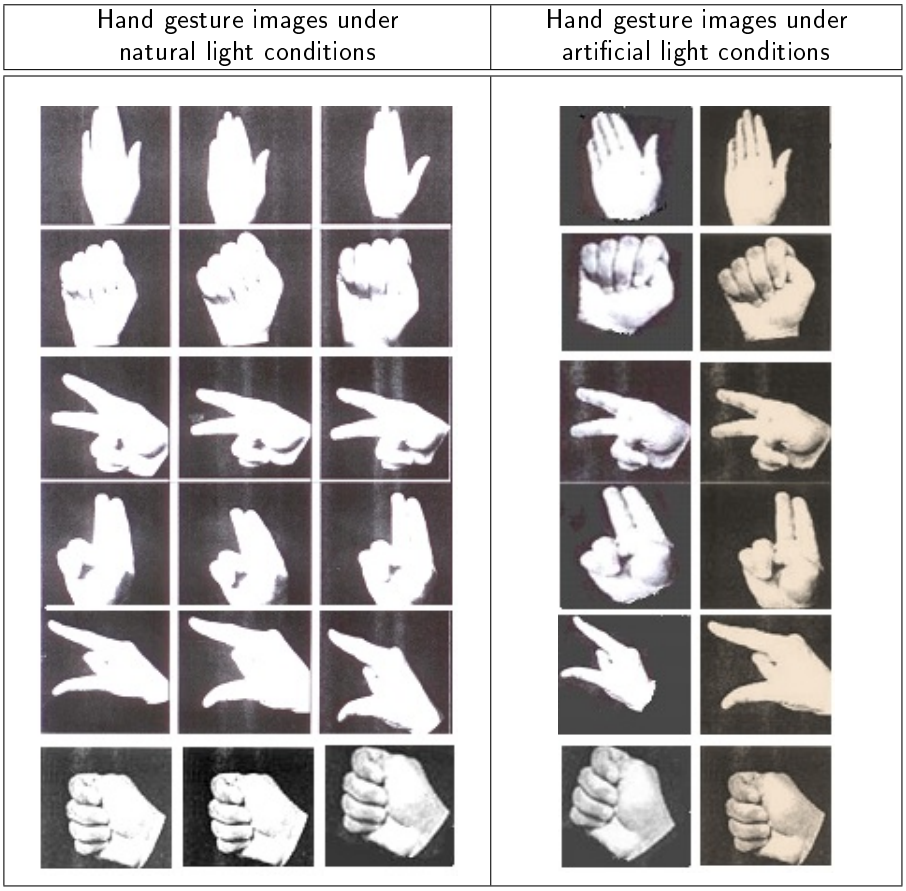

Figure 2.5: Training phase.

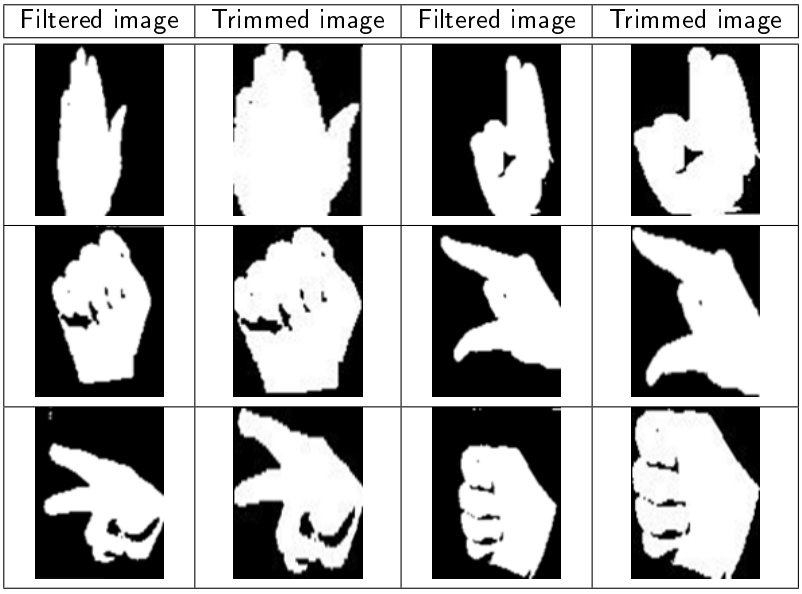

Figure 2.6: Image trimming effects.

2.4.3.1 Image Trimming Effect

The filtered hand gesture image may contain unused space surrounding the hand gesture. Thus, image trimming process is used to extract the hand gesture from its background. The effect of this process is shown in Fig.(2.6).

2.4.3.2 Coordinate Normalization

After scaling each image to a fixed size (250×250), the coordinates for the hand image are normalized between $[-1, +1]$.

2.4.3.3 Complex Moments Calculation

Each hand gesture in the training set will have a feature vector with 10 values, which represent the complex moments starting with zero order up to nine orders.

2.4.3.4 Training Phase

After the computation of feature vectors, each one (feature vector) contains 10 translation, scaling, and rotation-invariant elements characterizing the complex moments for the hand gestures. Five similar neural network classifiers are trained with a data set containing 30 feature vectors (training data set). These vectors were computed from the training set that includes five examples for each hand gesture performed by one subject. The learning process for the back-propagation neural networks is accomplished by using the parameters shown in Table 2.2.

Table 2.2: Parameters of Back-Propagation Neural Networks.

Parameters	Values
Input Layer	10 nodes
Hidden Layer	6 nodes
Output layer	6
Stop error	0.01
Learning rate	0.9

2.4.3.5 Testing Phase

After training the five neural networks using the training data consisting of 30 images, the performance is evaluated by applying the testing set on the network inputs and then computing the classification error. The testing process is conducted in the same manner as in the previous method. In this phase, 84 hand gesture images are used to test the system. Each one of the six hand gestures has a number of samples under different light conditions and with effects of scaling translation and rotation.

2.4.4 Summary

The experimental work has been carried out using the following setting regarding the data set and the parameters of neural network:

For hand contour method, the database consists of 30 images for the training set (five samples for each gesture) and 56 images for testing under different light conditions and with effects of scaling and translation. The parameters of the multi-layer perceptron neural networks for this method are the following: 1060 nodes for input layer, 100 nodes for hidden layer and 6 nodes for output layer.

For complex moments, the database consists of 30 images for the training set and 84 hand gesture images for testing under different light conditions and with effects of scaling translation and rotation. The parameters of the multi-layer perceptron neural networks for this method are the following: 10 nodes for input layer, 6 nodes for hidden layer and 6 for output layer.

2.5 Conclusions and Suggestions for Future Work

This research addresses the problem of static hand gesture recognition, and specifically, as the one of objectives of this study stated, develops a neural-based recognition system to recognize six selected static hand gestures (Open, Close, Cut, Paste, Maximize, and Minimize). The primary idea is to employ two feature extraction methods, namely, the hand contour method and the complex moments method, in the extraction of the features that characterize these hand gestures. These features are then used in training the neural networks to classify each gesture image into its respective class.

2.5.1 Conclusions

The primary conclusions can be summarized in the following points:

1. Hand contour-based neural networks training is evidently faster than complex moments-based neural networks training (at least 4 times faster where hand contour-based neural network took roughly between 110 and 160 epochs to converge, whereas complex moment-based neural network required at least between 450 and 900 epochs to convergence). This suggests that the hand contour method is more suitable than the complex moments method to real-world applications that need faster training, such as online training systems.

2. On the other hand, complex moments-based neural networks (86.90%) proved to be more accurate than hand contour-based neural networks (71.30%). In addition, the complex moments-based neural networks are shown to be resistant to scaling (96.15%) and translation (100%), and to some extent to rotation (65.38%) in some gestures (for example:open (100%), Maximum(80%)).The results indicate that the complex moments method is preferred to the hand contour method because of its superiority in terms of accuracy especially for applications where training speed is not very crucial, such as off-line training applications and desktop applications.

3. Hand contour features are less "distinguishable" compared to complex moments features. The high number of "not recognized" cases predicted via the hand contour method makes this evident (11.90% of the testing cases for hand contour against 1.20% for complex moments). The recognized class cannot be uniquely defined because there are two or more classes (gestures) that have the same high certainty (or probability) value.

4. Neural networks are powerful classifier systems, but they suffer from the problem of overfitting, a problem which was more visible with hand contour method. Less overfitting was observed with the complex methods method, which is considered as an advantage for this method as the learning techniques which avoid the overfitting problem can provide a more realistic evaluation about their future performance based on the training results. In addition, neural networks appear to be more efficient when the number of features or the dimension of the feature vector, which is equal to the number of nodes in the input layer, is moderate (e.g., the complex moments method with 10 nodes is more accurate than the hand contour method with 1060 nodes).

5. The current research aims to provide a generic system that can be customized according to the needs of the user by using each of the six gestures as a specific command. For example, a direct application of the current system is to use it as virtual mouse that has six basic functions, namely, Open, Close, Cut, Paste, Maximize, and Minimize.

6. In addition, the proposed system is flexible; it can be expanded by adding new gestures or reduced by deleting some gestures. For example, you can use four gestures for TV control application, with each gesture being translated into one

TV command: "Open": to turn on the TV; Close: to turn off the TV; Min: to reduce the sound volume; and Max: to increase the sound volume, and so on.

2.5.2 Suggestions for Future Works

Possible ways to extend and improve this work are suggested below:

1. Although the neural networks methods have been widely recognized as powerful classifier methods, other classifiers, such as the Hidden Markov Model (HMM) or the Support Vector Machine (SVM), may also be used for this problem along with the two feature extraction methods.

2. One possible way to reduce the "not recognized" cases in the gesture recognition process is to employ ensemble classifiers, where the members of the ensemble are various types of classifiers such as decision trees, fuzzy systems, SVMs, etc. The recognized gesture, in this case, is the one which receives the highest number of votes from the ensemble members.

3. The two feature extraction methods employed in this study can be applied for other hand gesture-based applications. In this case, the current system should be adjusted to fit the new application by, for example, changing the number of neurons in the output layer of ANNs to correspond to the number of gestures to that the system needs to recognize.

References

[1] Y.S. Abu-Mostafa and D. Psaltis. Recognitive aspects of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.PAMI-6(6):698–706, 1984.

[2] M.A. Amin and H. Yan. Sign language finger alphabet recognition from Gabor-PCA representation of hand gestures. In *International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 4, pages 2218–2223, 2007.

[3] V. Athitsos and S. Sclaroff. Boosting nearest neighbor classifiers for multiclass recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 45, 2005.

[4] G.J. Awcock and T. Awcock. *Applied Image Processing*. McGraw-Hill, 1995.

[5] C. Bekir. Hand gesture recognition. Master's thesis, Dokuz Eylul University, 2012.

[6] C.C. Chang, J.J. Chen, W.K. Tai, and C.C. Han. New approach for static gesture recognition. *Journal of Information Science and Engineering*, 22(5):1047–1057, 2006.

[7] G. Dong, Y. Yan, and M. Xie. Vision-based hand gesture recognition for human-vehicle interaction. In *International conference on Control, Automation and Computer Vision*, volume 1, pages 151–1551, 1998.

[8] L.V. Fausett. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall Englewood Cliffs, 1994.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, page 178, 2004.

[10] W.T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *IEEE International Workshop on Automatic Face- and Gesture- Recognition*, pages 296–301, 1995.

[11] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition, 2002.

[12] S. Gupta, J. Jaafar, and W.F.W. Ahmad. Static hand gesture recognition using local Gabor filter. *Procedia Engineering*, 41:827–832, 2012.

[13] C. Hu, M.Q. Meng, P.X. Liu, and X. Wang. Visual gesture recognition for human-machine interface of robot teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1560–1565, 2003.

[14] D.Y. Huang, W.C. Hu, and S.H. Chang. Vision-based hand gesture recognition using PCA+Gabor filters and SVM. In *5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pages 1–4, 2009.

[15] J. Huang. *Color-spatial image indexing and applications*. PhD thesis, Cornell University, 1998.

[16] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu. Dynamic hand gesture recognition using the skeleton of the hand. *EURASIP Journal on Applied Signal Processing*, 13:2101–2109, 2005.

[17] A. Just. Two-handed gestures for human-computer interaction. Technical Report 6(73), IDIAP Research Institute, 2006.

[18] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2479, 2010.

[19] R.Z. Khan and N.A. Ibraheem. Hand gesture recognition : a literature. *International Journal of Artificial Intelligence & Applications*, 3(4):161–174, 2012.

[20] T.K. Kim and R. Cipolla. Gesture recognition under small sample size. In *8th Asian Conference on Computer Vision (ACCV)*, volume 4843 of *Lecture Notes in Computer Science*, pages 335–344, 2007.

[21] M.W. Krueger. *Artificial Reality II*. Addison-Wesley, 1991.

[22] A. Licsar and T. Sziranyi. Hand-gesture based film restoration. In *2nd International Workshop on Pattern Recognition in Information Systems (PRIS)*, pages 95–103, 2002.

[23] R.K. McConnell. Method of and apparatus for pattern recognition, 1986.

[24] H. Meng, F. Shen, and J. Zhao. Hidden Markov models based dynamic hand gesture recognition with incremental learning method. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3108–3115, 2014.

[25] A.K. Musa. Signature recognition and verification by using complex-moments characteristics. Master's thesis, University of Baghdad, 1998.

[26] S. Oprisescu, C. Rasche, and Su Bochao. Automatic static hand gesture recognition using ToF cameras. In *20th European Signal Processing Conference (EUSIPCO)*, pages 2748–2751, 2012.

[27] F. Parvini and C. Shahabi. An algorithmic approach for static and dynamic gesture recognition utilising mechanical and biomechanical characteristics. *International Journal of Bioinformatics Research and Applications*, 3(1):4–23, 2007.

[28] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[29] P.D. Picton. *Neural Networks*. Palgrave Macmillan, 2nd edition, 2000.

[30] I. Pitas. *Digital Image Processing Algorithms and Applications*. Wiley-Interscience, 2000.

[31] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust part-based hand gesture recognition using Kinect sensor. *IEEE Transactions on Multimedia*, 15(5):1110–1120, 2013.

[32] V.D. Shet, V.S.N. Prasad, A.M. Elgammal, Y. Yacoob, and L.S. Davis. Multi-cue exemplar-based nonparametric model for gesture recognition. In *Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, pages 656–662, 2004.

[33] B. Swapna, F. Pravin, and V.D. Rajiv. Hand gesture recognition system for numbers using thresholding. In *1st International Conference on Computational Intelligence and Information Technology (CIIT)*, volume 250 of *Communications in Computer and Information Science*, pages 782–786. Springer, 2011.

[34] K. Symeonidis. Hand gesture recognition using neural networks. Master's thesis, School of Electronic and Electrical Engineering, Centre for Vision, Speech and Signal Processing, Surrey University., 2000.

[35] J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex backgrounds. In *2nd International Conference on Automatic Face and Gesture Recognition (AFGR)*, pages 170–175, 1996.

[36] S.E. Umbaugh. *Computer Vision and Image Processing: A Practical Approach Using Cviptools*. Prentice Hall, 1997.

[37] R.-L. Vieriu, B. Goras, and L. Goras. On HMM static hand gesture recognition. In *10th International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 1–4, 2011.

[38] Y. Wu and T. S. Huang. *Gesture-Based Communication in Human-Computer Interaction*, volume 1739 of *LNCS*, chapter Vision-based gesture recognition: a review, pages 103–115. 1999.

[39] L. Yun and Z. Peng. An automatic hand gesture recognition system based on Viola-Jones method and SVMs. In *2nd International Workshop on Computer Science and Engineering (WCSE)*, volume 2, pages 72–76, 2009.