CHAPTER 1
INTRODUCTION

This section will describe the handwriting recognition system with more details in regarding operation modes and the constraints available in such systems.

1.1 Handwriting Recognition Systems (HR Recognition)

Handwriting recognition system can be defined as a process that is applied on a row of handwritten letters in order to recognize them. This process varied and enhanced in decades and yet this field is wide open for a researcher for more novel ideas to be investigated.

Handwriting recognition technology started by inventing Optical - based systems, which scans the handwritten text and recognize it. Other systems were invented as a Pen – based ones, where the system tracks the pen tip and matches the written information. Although the improvements and the enhancements in this field are stepping up, the accuracy of these systems still make them less user friendly.

The importance of the Handwriting recognition systems is increased time by time not only as a key tool for recognizing a row of handwritten text of data, but also as a possibility to replace the PC's mouse, drawing scenes, identity verifications and more of the future uses.

As mentioned earlier, handwriting technology available in the market is far from satisfaction and the wheel of improvement is in progress. In this thesis, we introduce a novel idea that reduces the constraints related to handwriting recognition which is based on processing row of images that describe the movements of the hand's writer to formulate the written letters.

1.2 Online & Offline operation mode

There are two modes for handwriting character recognition, off-line and on-line. In off-line, which is the technology applied in Optical – based systems, user is prompted to write a complete page of words and then a camera or a scanner is used to take a raster image of data. This technique extracts static information form an image where there is no relationship between image pixels and writing direction or the writing order of the strokes. Whereas, in on-line systems, which is the technology applied in Pen – based systems, extracted dynamic information represents the pen-down state movement of a pen digital tablet. During writing process, real time x,y samples represent writing direction and writing order of the strokes[1].

1.3 Recognition System User Dependence

One of the important categorization in character recognition systems is how the system depends on the user writing style. The system is called user-independent when recognition system can handle multiple users writing styles, whereas a user-dependent system is adapted and trained to recognize a single user. One of the constraints that affect the selection of such classes is that each character belonging to a single user could have variability within writing the character itself. Therefore, a user-independent system requires large a number of samples which impact computational cost and recognition accuracy. It should be noted that many handwriting devices are meant to be used by a single user, hence user-independent systems are the right choice for such devices [1].

1.4 Constraints on Arabic Handwriting Style

Arabic letters have unique characteristics that make the recognition process a challenging task. These characteristics include letter connectivity, dots effects, overlapping property and variability in shapes and fonts.

1.4.1 Letter Connectivity

Arabic words are formed by connecting multiple letters altogether. However, some of the letters are not connected and spaced to form one word as shown in figure 1-1. As a result, we need to have a system that can distinguish between one letter and different letters form just one pen movement.

Figure 1-1 One motion includes 3 letters

1.4.2 Dots effects

Many letters in Arabic handwriting are distinguished by variable number of dots. As a result, the 28 Arabic letters are reduced into 15 primary shapes as shown in the figure 1-2.

Figure 1-2 15 Primary letters

With the existence of such constraint, faulty recognition will happen if dots are misplaced. Figure 1-3 shows that (ي) could be recognized as (ب) if one of the lower dots is missed or to make it worse another possible upper dot from the letter (ق) shifted to letter (ي) and wrong recognition will occur.

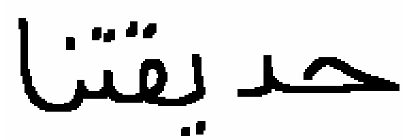

Figure 1-3 Example on dotting ambiguity

1.4.3 Overlapping property

Letters can overlap which increases the importance of developing a segmentation methodology that is able to segment cursive words without adding ambiguities to the letter itself. Figure 1-4 shows samples of such constraints where letters (م) and (ل) are overlapped and we have to segment them without additional ambiguities that could lead to a faulty recognition.

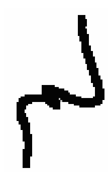

Figure 1-4 Overlapped letters

1.4.4 Shape Variability

There is a group of Arabic letters that have different shapes depending on where the letter is located in the word. In figure 1-5 we can notice two versions of letter (ك) which means we have two main shapes for this letter which could increase the recognition complexity. In order to avoid such constraint, we have to develop accurate classifier that handles variable letter shapes.

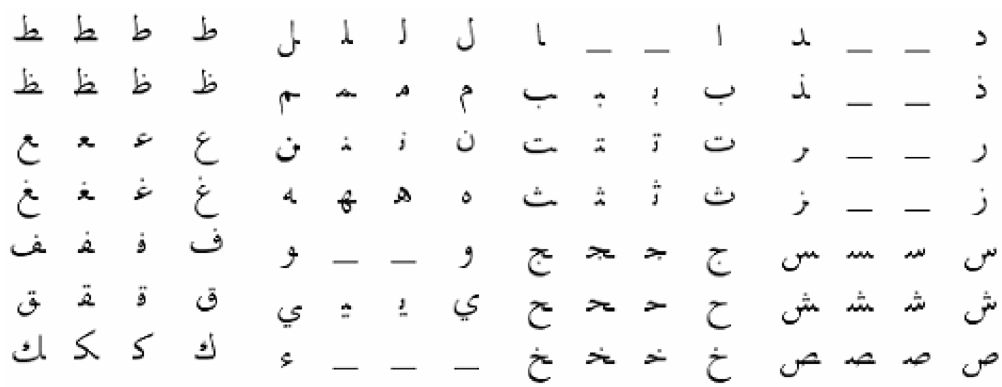

Figure 1-5 Different letter shapes for different locations in the word

1.4.5 Fonts Variability

There are many properties that complicate Arabic handwriting recognition such as the font type with different writing styles. Variability for one word with different font types is shown in figure 1-6, where letter (س) can't be recognized even as a letter in Roq'aa font type.

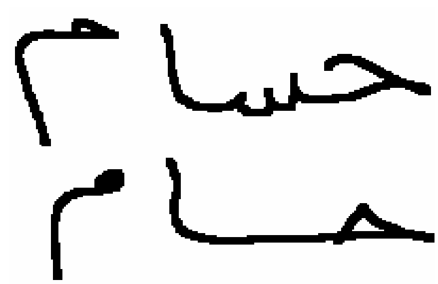

Figure 1-6 Two different Fonts, Traditional and Roq'aa

1.5 On-line Arabic Handwriting Review

On-line Arabic handwriting recognition is a relatively new area that researchers started to tackle since early 90’s. This section will summarize the major work done in this field.

- Hierarchical Rule-based Approach

This approach uses hierarchical rule-based classifier which uses the number of strokes forming the handwritten letter. The criterion of classification is based on segmenting the letter into 4 groups (initial, medial, final and isolated) increasing the classification accuracy. [21].

- Segmented Structural Analysis

This approach uses decision tree classifier applied on segmented Arabic postal address words. The criterion used is finding extreme curvature for each isolated letter. [2].

- Structural and Fuzzy Approach

The classifier described in this approach uses structural analysis to find out each latter class and uses fuzzy logic analysis to allow wide variation in people’s handwriting within the same class [3]

- Template matching and Dynamic Programming

In this approach, each letter's class is identified by a pre – specified prototype using Template matching analysis. The classification is done using dynamic programming classifier which minimizes the distance between the prototype and input letter [4].

- Hierarchical Template Matching and K- NN Classification

This approach uses template matching methodology in order to learn letter’s features like number of dots, their position with respect to the major strokes, number of minor strokes and their slope. Then, a KNN classifier is used to find out the closest class by encoding the main strokes as a primitive of angular directions [5].

- Neuro-Fuzzy

In this approach, the criterion of defining the letter features depends on number of strokes. Each letter is represented in 6 strokes at most to have 6 feature vectors. These vectors were given to a fuzzy beta radial basis function neural network to define each latter's class. [6].

- Pruned Kohonen Maps

Combining Kohonen maps trained on Fourier descriptors and tangent vectors. Classification is done by how each letter match the closest map [7].

- Neural Network

In this approach, Self-Organizing Maps (SOM) are used for feature extraction and Perceptron neural network is used as a classifier [8].

Table 1-1 shows accuracy level of each method compared to others and what data sets are used in training and testing.

Table 1-1 Results of the Previous work

Approach	Dataset	Accuracy
Hierarchical Rule-based	Large variable-forms of Arabic dataset	100%, it dropped heavily with an ambiguity in strokes numbers.
Segmented Structural Analysis	10 writers with a set of 120 postal code words each of 13 characters.	Good accuracy, but sensitive to rotation. 100% when the user instructed to modify his writing to avoid the weakness.
Structural and Fuzzy	Have datasets for training	No test shown but had perfect training results.
Template matching and Dynamic Programming	9 prototypes on test data for one user.	96% only used one test subject who varied his handwriting across the prototypes.
Hierarchical Template	Testing 7 writers on sets of 60 characters.	84%, increased into 93% after Weighting the features

Matching and K- NN Classification		manually depending on the previous test accuracy.
Neuron-Fuzzy	One word and a small subset of 7 different letters.	89% without dot and diacritical information
Pruned Kohonen Maps	7344 samples of 17 classes written by 17 writers.	93.54%
Neural Network	Multiple unconstrained 19 writers.	Recognition rates above 80% and training recognition rates above 90%.

1.6 Video Based Handwriting Recognition Systems Review

Arabic on-line handwriting recognition systems based on text processing shows high recognition rates at the expense of user-friendliness, where these systems reach optimal performance when the user is restricted in his writing style. In order to lower the effect of using this trade off, video images are introduced in the recognition process, where these images describe the hand movements of the writer.

In this proposal, we extract the dynamic information of the hand movements by applying the temporal analysis which is concerned in accumulating the image differences in order to store the traces of the hand movements that will describe the written letter, and then by applying the spatial analysis like principle components, discrete cosine transform, and accumulated differences in projection and Walsh – Hadamard Transform.

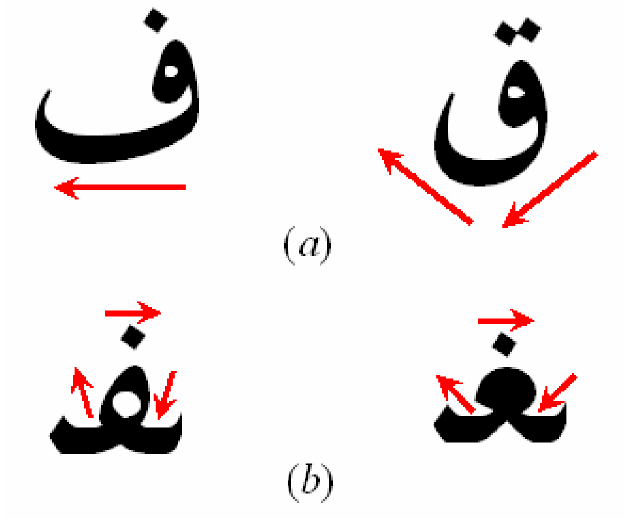

Figure 1-7 Video processing enhances the recognition process

Different methods were proposed by researchers, and the majority of them were using the pen tip tracking method. Video based methods that depend on the pen tracks suffer from low recognition rates since following an object in image sequence is an easy way to have faulty classifications. Furthermore, the recognition rates are highly dependable on the camera data capturing rates which is another major factor which will reduce the success of classification results. [9].

When we try to track every pixel at high speed writing styles, captured images will fail to recover all the detailed strokes due to incorrect pixel connections. [10]. Another critical factor that affects how the letters are recognized is the image resolution which must be accurate enough to facilitate the single strokes and their smaller portions. Another problem could arise from using this method is that the strokes generated when the pen tip is lifted off the paper will capture redundant hand movements that will totally change the letter shape. In addition, delayed strokes can be generated in writing letters like (t) or (I) where the pen will back a distance as it will add unnecessary traces [9]. Another important factor is that the pen traces could be mixed with pen tip, hand and their shadows [10].

An approach developed by Munich & Perona in 1996 was to capture handwritten text with an ordinary pen on a paper [9]. However, this approach still suffers of low recognition rates due to tracking pen traces.

Another approach is proposed to track the ink pixels resulted from pen writing instead of tracking the pen tip itself and uses the difference in images to obtain the ink trace left on the paper [9]. , but this approach still finds a way for redundant details in the image like pen, hand movements and shadows. [10].

There is also a comparable approach that models a specific letter using chain codes to recognize the text data and uses the last image frame from video camera to enhance the recognition process. Chain code represents the stroke direction and sums of all the chains' direction together to have the final form of the letter. The video part of this approach segments the last video frame into different size boxes where each box corresponds to a letter. The quality of this approach degrades for characters of similar chains like 5, c and C or u and v. Moreover, the system is restricted to a limited number of letters and symbols [11].

Another approach developed for Chinese alphabet extracts stroke dynamic information from video sequence frames in two steps. Initially, if tracks both run direction and strokes writing order and then processes the last frame (which includes the full strokes of the letter). This method looks promising but the errors are in the run direction estimation which will generate incorrect letters [9].

All of the above methods use normal paper, but another proposed one used the white board with video camera. In this approach, off-line recognition is used to extract text from the full view image by dividing it into several blocks. Each one is used to extract the strokes, pixel counter and the pixel intensities [12]. The main problem of this method is its sensitivity to noise which yields reduced accuracy.

1.7 Thesis Organization

This thesis is organized into many chapters. It starts with an overview of the proposed online handwriting recognition system in chapter 2. In chapter 3, there is a full description of the database used in our handwriting system in Database description. This chapter focuses on the system settings, surrounding environment and details on the key hand movements that are critical for the system performance. Feature Extraction in chapter 4 explains the novel method that is developed in this thesis to enhance the Arabic handwriting recognition. This chapter focuses on temporal and spatial analysis being used and how different schemes are developed to extract the features from row images. Classification in chapter 5 explains how KNN and HMM classifiers are being used and the Experimental Results in chapter 6 show the performance of using these classifiers consecutively. The last sections are Conclusions and References.

CHAPTER 2

METHODOLOGY

A Novel Method is proposed to recognize online video based Arabic handwriting letters by extracting dynamic information from row images that describe the hand movements for a specific letter. This information is captured via video camera setup that focuses in the writer's hand movements as shown in figure 2-1.

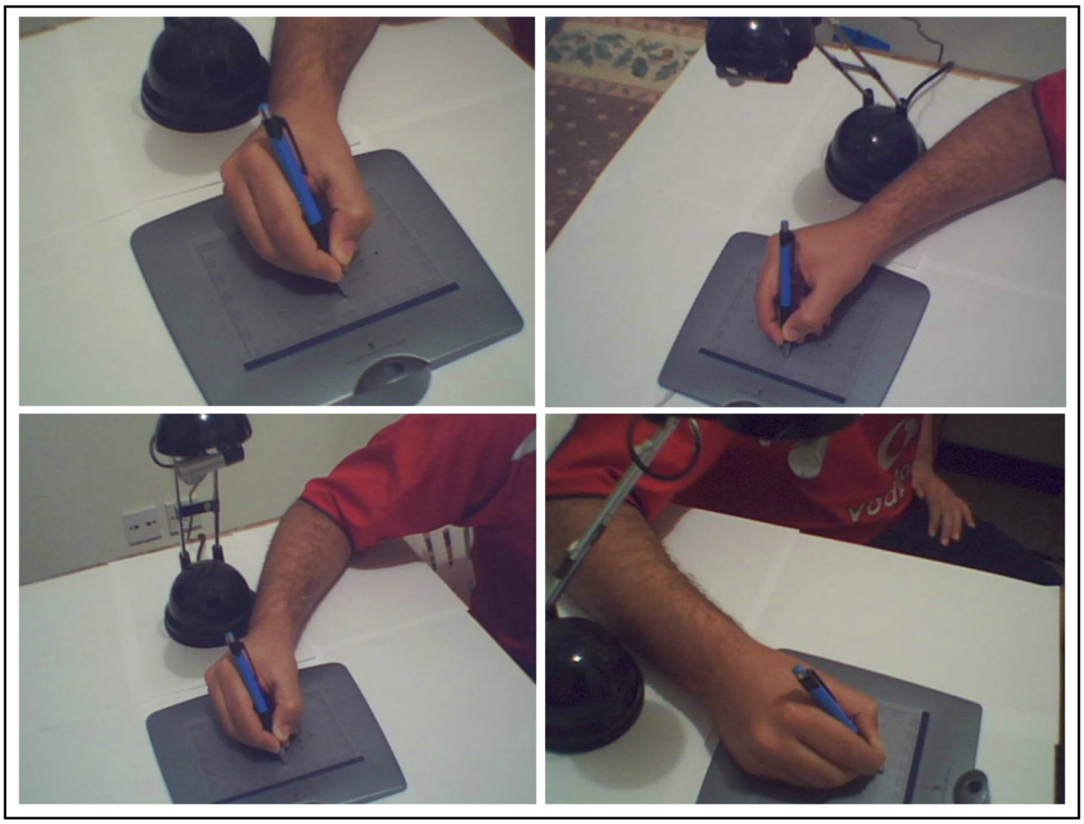

Figure 2-1 System overviews

The capturing process is achieved in predefined settings that take into consideration the video settings and the inevitable surrounding conditions as lightings and camera's position. The hand movements for each letter are unique and distinguishable, for this reason, these movements are analyzed and categorized depending on the main writing style of each letter. For example, letters like (ب ث ن ت) all are having a common flatted – base writing shape as shown in figure 2-2.

This main shape is a key factor in distinguishing between each letter and the number of dots yields more quality in classification.

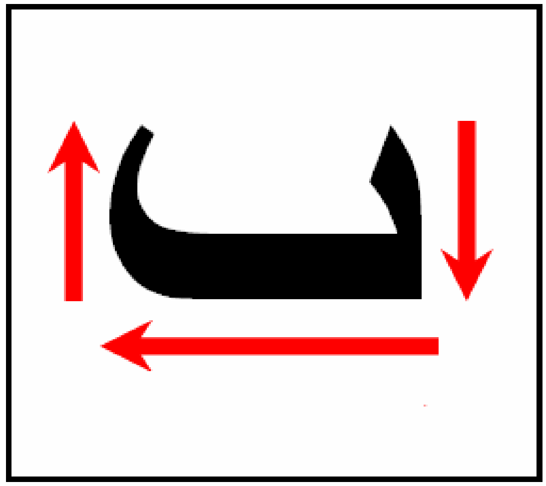

Figure 2-2 Flatted – based letters

Now, after the system is installed and the letter hand movements are designated, we need to extract dynamic information stored in the captured row images as shown in figure 2-3. In this thesis, we apply temporal and spatial analysis to form a feature vector that describes hand moves in image row. Temporal analysis is applied to store the accumulated differences of row images by using several proposed schemes. Each scheme has a unique methodology in how its "Looking for" dynamic information is stored in the moving pictures. Further reshaping analysis is applied on these schemes using spatial analysis techniques like DCT and Walsh – Hadamard transform.

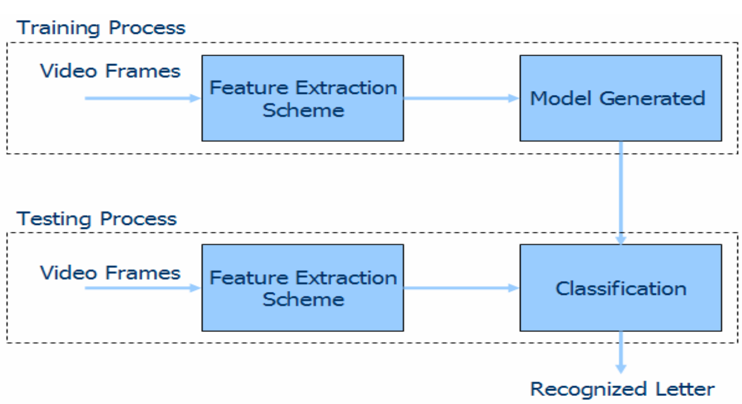

Figure 2-3 On-line Video-based handwriting recognition system

The handwritten letters are then recognized and classified using either K Nearest Niehber (KNN) or Hidden Markov Model (HMM) classifier. In the former classifier, the distance between the test letter and the training ones is measured and compared to all the other letters. Once we found the minimum one, we could easily classify the letter under testing process. In the latter classifier, a model of the letter with the right number of states is identified, and then this model is used to compare it with testing letters and find the best letter match.

CHAPTER 3

DATABASE COLLECTION

This section describes the system setup and the setting applied for video capturing. It also describes a full analysis of hand movements that characterize each handwritten letter.

3.1 Instrumentation

Details on the system setup are described below.

3.1.1 System Setup

Our handwriting recognition system uses a CREATIVE Laptop video camera which is installed in appropriate height to GENUIS handwriting pressure sensitive tablet as shown in the figure 3-1. The Laptop video camera is suspended on a camera handler that has variable height sizes and variable arc distances. All these mechanical options are helpful in zooming and positioning the camera.

Figure 3-1 System's Setup

The laptop video camera is installed in appropriate height of 15 cm that covers the whole tablet which is enough to capture all the detailed hand movements within area of 36 cm^2 during the writing process. Also we took into consideration the lighting effects where the system is located in such a way to avoid any probable hand shadows. Moreover, the light must be enough so that no more constraints are added to the system that could affect the recognition performance.

3.1.2 Video Settings

The following are details of the setting being used in capturing hand movements during the handwriting process.

3.1.2.1 Capturing Resolution:

The row images are captured at resolution of 120x160 pixels, which is enough to show the details of the hand movements without displaying the images in uncorrupted frames. The resolution selected is a bit small in order to increase the computation of speed and reduces the complexity of the analysis.

3.1.2.2 Frame Rate:

The frame rate being used is 8 frames per second for each Arabic letter. It’s selected in order to be fast enough to guarantee the full hand movements that describe the required letter. In the other hand, the hand movement’s speed applied to write a letter sample must be enough to ensure that there is no missing moves nor extra zero information images which will result in misclassification later on.

All of the written letters are covered in a total number of 16 frames. Most of the letter main pattern is captured in the first 50% of the frames and the rest 50% includes the unique details like dots, special characters and distinct hand movements.

Figure 3-2 shows that the hand movements for tracing a letter’s dot are a vital issue to differentiate between letter (ب) and letter (ن) while special character (ء) is definitely a key to differentiate between letter (ل) and letter (ك) although both have the same hand movements' traces.

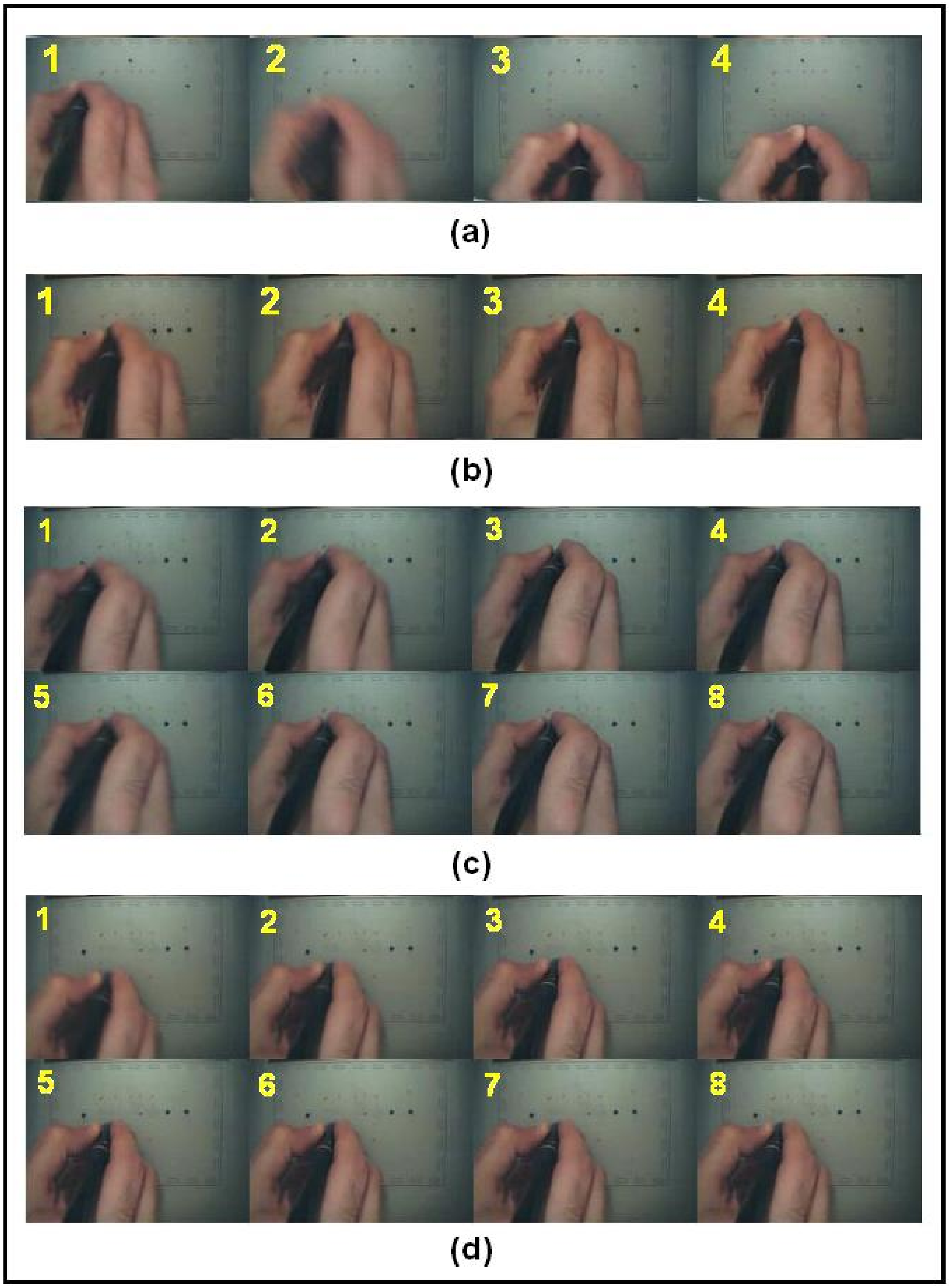

Figure 3-2 Unique frames captured from the hand movements, which considered being vital in the letter recognition

3.1.2.3 Triggering Rate:

The time between current and next frame is 0.125 sec. This time is constant so that no interrupted moves during writing a letter sample could cause a changes in the letter patterns.

3.2.3 Capturing Conditions

3.2.3.1 Lighting:

Lighting is an important factor that must be taken into consideration during the writing process. It needs a fine control in order to avoid noise effects such as shadows, over-lighting or under-lighting.

3.2.3.2 Camera's Position:

The position of the video camera affects the captured images, especially in zooming case or could add more redundant components when the writer's hand is out of position. Camera's position describes the appropriate height for the camera which is installed on a well curved handler with appropriate zooming option on the moving hand.

3.2.3.3 Writing Style:

The style of writing used in this thesis is the standard Arabic handwriting font, which has unique characteristics that distinguish each letter from the others. The user is ordered to use the same style for all letters to avoid any noise patterns that could result in misclassifications. Arabic letters used in this thesis are written in isolated form.

3.2.3.4 Database size:

Two different users used the system. Each one wrote 40 samples per letter which are collected in similar circumstances and conditions. Each letter samples is composed of 16 frames per letter.

3.2 Letter’s Movements Analysis

Arabic letters are characterized by unique handwriting movements compared to other handwriting languages. In this section we will focus on the analysis of these handwriting movements that form the main pattern of each letter. This analysis is very useful in our system development. In this thesis, we have noticed 11 unique patterns for the 28 letters that the Arabic language is composed of.

3.2.1 Flattened – Base movements

One of these unique patterns is similar to a Flattened – Base style, where the letter main pattern is based on 3 hand movements. Figure 3-3 shows how letters like (ب ت ث ن) have flattened – base writing shape. However, the existence of letter’s dots with variable numbers and locations is very helpful in letter recognition.

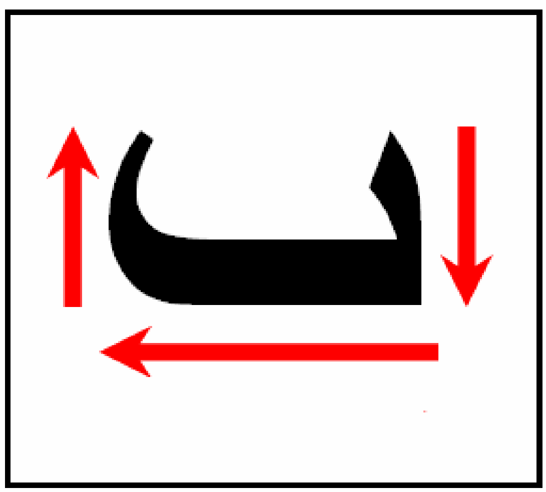

Figure 3-3 Flattened – based letters

3.2.2 Half - Circular movements

Another style of writing is half – Circular where the hand moves as shown in figure 3-4 and the letters belonging to this style are (خ ج ح غ ع). We can easily recognize this group among other letters by noticing how the first three start with straight segment in addition to Half - Circular moves compared to the rest two with a small circle at the initial frames of writing. These unique moves and different dot's locations are all key information in letter recognition.

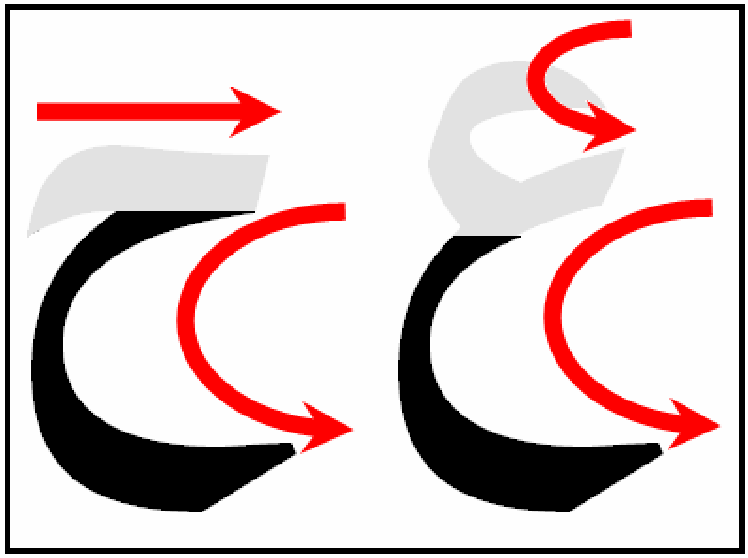

Figure 3-4 Half – Circular movements

3.2.3 Narrow - Angled movements

In this group of letters, the pattern starts with the handwriting in an angle less than 90° as shown in figure 3-5. this pattern along with the available dots yield high recognition rates.

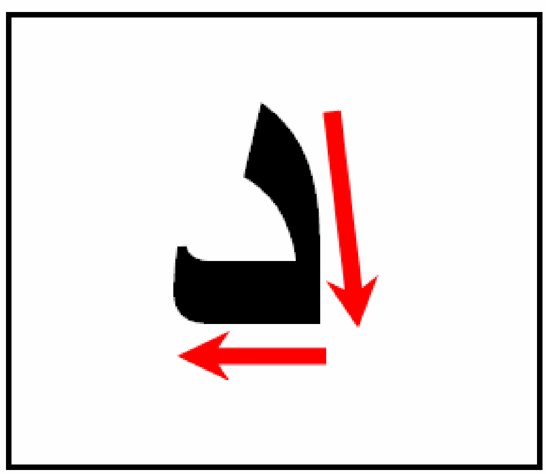

Figure 3-5 Narrow – Angled movements

3.2.4 Elliptic – Started movements

Another group of letters having unique hand movements which are helpful in recognition is (ظ ط ص ض). This group is characterized by elliptic started moves appeared in the first frames and half circular moves or straight segment appeared in the rest ones as shown in figure 3-6.

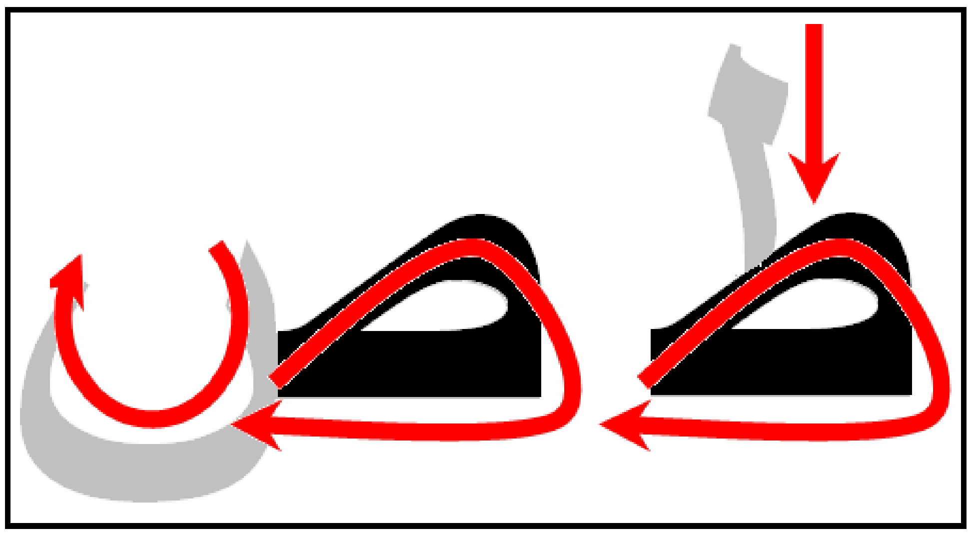

Figure 3-6 Elliptic – Started movements

3.2.5 Saw-toothed movements

Letters like (ش س) have Saw-toothed way of writing as shown in figure 3-7. These movements are not available in the other writing styles which increase the recognition rates without forgetting the three dots that differentiate between both letters.

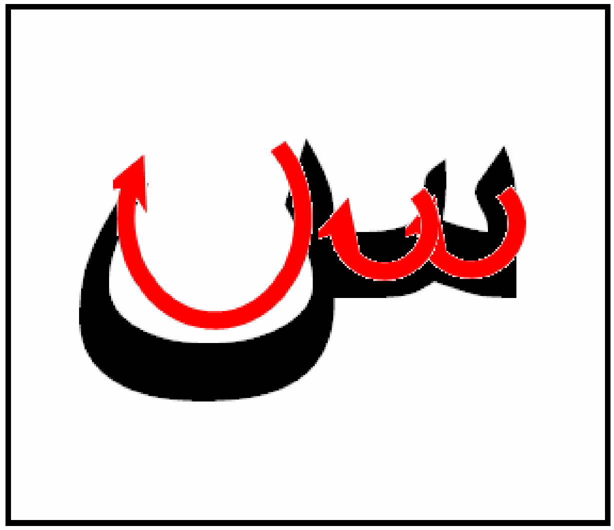

Figure 3-7 Saw- Toothed movements

3.2.6 Arc movements

Letters like (و ز ر) have a unique curvature similar to an arc. This group can be easily distinguished among itself because of a dot in writing the letter (ز) and a small tiny circle before witting the letter (و) as shown in figure 3-8.

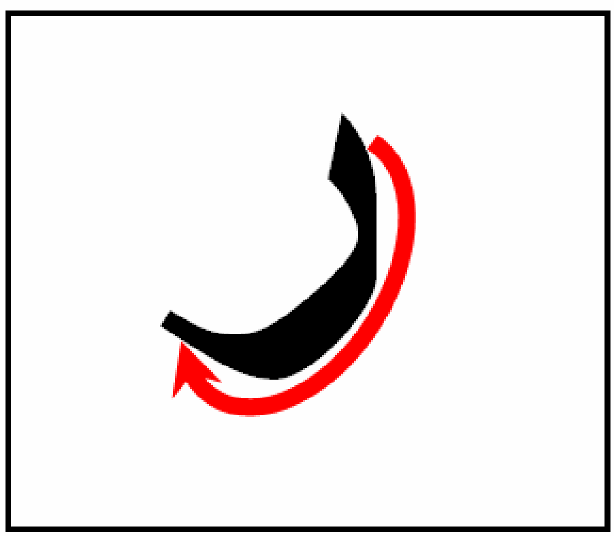

Figure 3-8 Arc movements

3.2.7 Linear movements

Writing letters in a linear shape can be found in (أ م) . Both have similarity at the end of the writing process, but at the initial frames, we have special features that specializes one letter among the other like (ء) and tiny half circle in (م) as shown in figure 3-9.

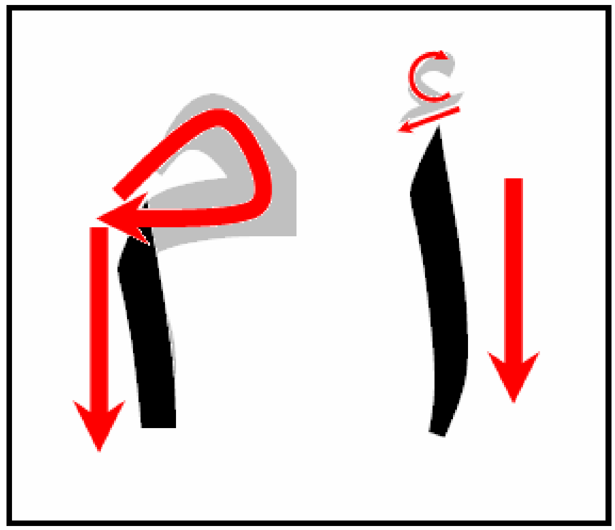

Figure 3-9 Linear movements

3.2.8 Flattened with Tiny circle movements

The style of writing letters in a shape of Flattened with tiny circle can be found in (ف ق) as shown in figure 3-10. This shows how Arabic handwriting letters have unique hand movements that increase the recognition rates.

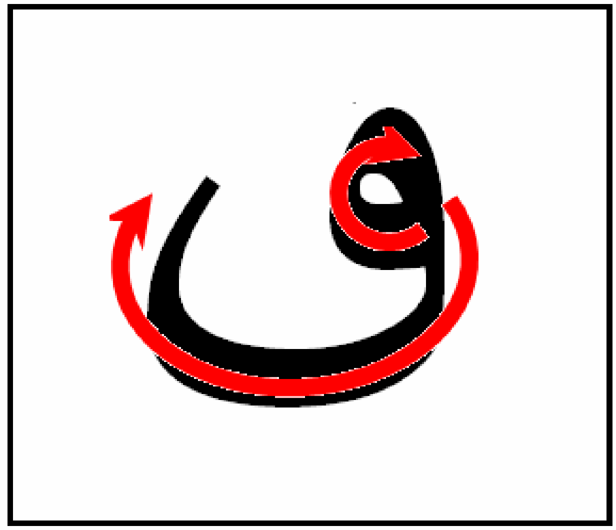

Figure 3-10 Flatted with Tiny circle movements

3.2.9 Reverse L movements

Letters like (ك ل) are almost having the same shape of writing similar to a reverse L as shown in figure 3-11 and a special character like (ء) which is useful in distinguishing between both letters.

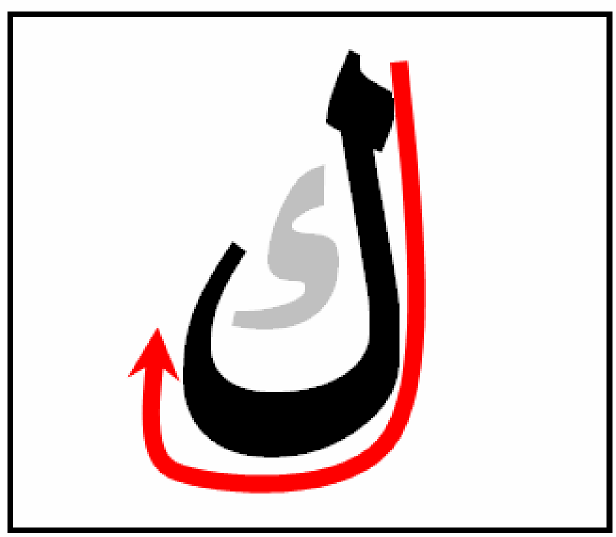

Figure 3-11 Reverse - L movements

3.2.10 Full Circular movements

Letter like (هـ) is composed of two full circles during the handwriting movements as shown in figure 3-12. This increases the letter recognition rates.

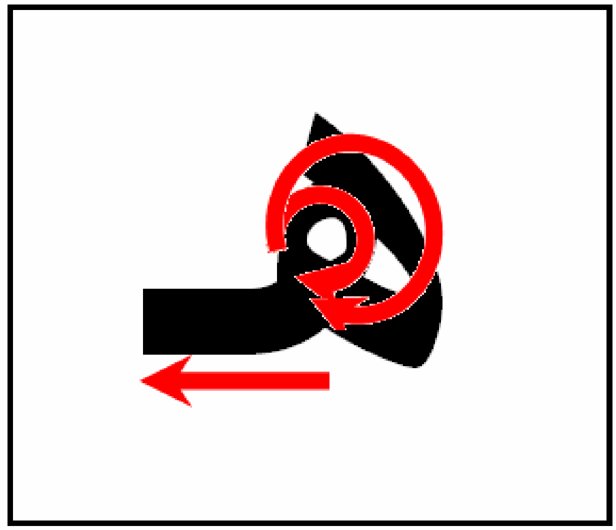

Figure 3-12 Full Circular movements

3.2.11 Multi - Curved movements

Letter like (ي) is also characterized with curved hand movements that dose not exist in any other Arabic letters as shown in figure 3-13**.**

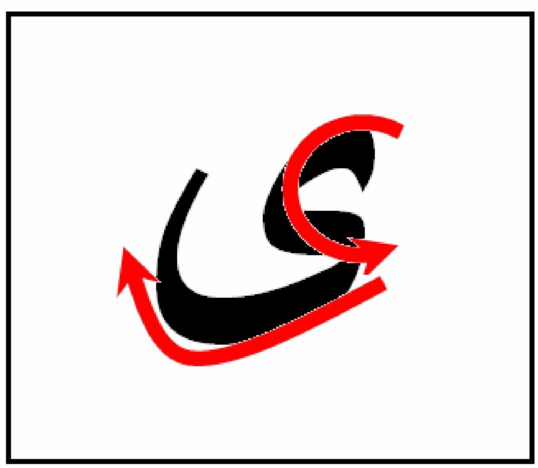

Figure 3-13 Multi – Curved movements

CHAPTER 4

FEATURE EXTRACTION

The feature extraction process used here is composed of two major steps: First, we use temporal analysis to capture the motion in the video sequence and convert it to one or two still images. Next, we apply spatial analysis to further compress the temporal analysis into a relatively small number of parameters.

4.1. Temporal Analysis

Image differences are an approach that compares images on pixel basis to detect the motion information. A difference image can be defined as

$$d_{ij}(x,y)=\begin{cases}1 & f(x,y,t_i)-f(x,y,t_j)>Th_{ij}\\ 0 & otherwise\end{cases} \qquad (4\text{-}1)$$

Where $d_{ij}(x,y)$ has a value of 1 at spatial coordinates (x,y) only if the gray level difference between the two images is greater than the threshold Th. Th can be defined as either 0 or the mean of moving pixels or one standard deviation above that mean. Thresholded image differences are then accumulated into one final image that captures the motion of the sequence. Accumulated Differences (AD) of an image sequence can be classified into three types: Absolute, Positive and Negative; formally:

$$|AD|(x,y)=\begin{cases}AD+1 \ if & |f(x,y,t_k)-f(x,y,t_{k-1})|\geq Th_{(k,k-1)}\\ AD & otherwise\end{cases} \qquad (4\text{-}2)$$

$$AD_+(x,y)=\begin{cases}AD_+ +1 \ if & (f(x,y,t_k)-f(x,y,t_{k-1}))\geq Th_{(k,k-1)}\\ AD_+ & otherwise\end{cases} \qquad (4\text{-}3)$$

$$AD_-(x,y)=\begin{cases}AD_- +1 \ if & (f(x,y,t_k)-f(x,y,t_{k-1}))\leq -Th_{(k,k-1)}\\ AD_- & otherwise\end{cases} \qquad (4\text{-}4)$$

Since many Arabic letters differ by having additional dots like (ح خ , د ذ , ع غ, ر ز and so), it makes sense to increase the weight of accumulated differences at the latter half of the image sequence where the additional dots are usually represented. This varying weight is represented by w_k in the 4 – (2-4) above.

Figure 4-1 shows a video sequence of the letter ث and its corresponding absolute, positive, and negative ADs. In case of Absolute ADs, it stores all hand movements as a pattern segment regardless of the motion direction as shown in figure 4-1b. In the positive ADs, the main body of the letter (which has the shape ں) is captured where it has a rather consistent motion direction from right to left. In the negative ADs, the movement of the hand is in the opposite direction where the hand moves back to write the letter dots. (Hint: the yellow arrows show the 3 dot for the letter)

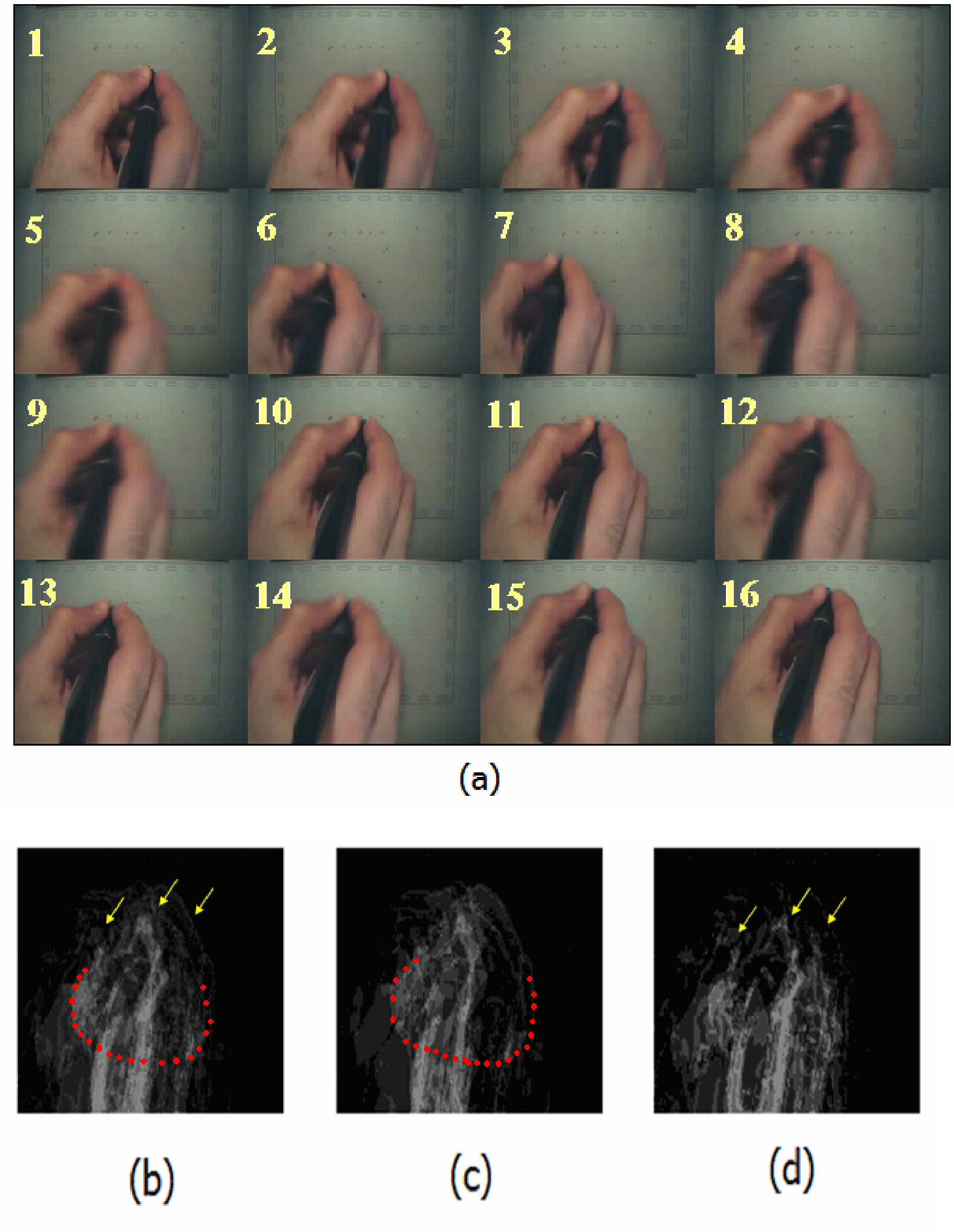

Figure 4-1 (a) Hand movements for letter ث, (b) Absolute AD, (c) Positive AD, (d) Negative AD

4.2 Spatial Analysis

4.2.1 Image alignment via principle component analysis (PCA)

To ensure that the extracted temporal features are rotational invariant (i.e insensitive to the direction of writing), PCA is applied to align the ADs prior to spatial feature extraction. The major principle of directions of the reference and the test AD images are aligned via PCA. The major principle component of a two-dimensional dataset defines a line that encapsulates the maximum amount of variances in that dataset. PCA is applied to an AD image after binarizing the image and treating the Cartesian coordinates of the "1" pixels as a two-dimensional data set **X.** The direction of the major principle component is that of the eigen-vector of the covariance matrix of **X** that corresponds to the largest eigen value[19].

4.2.2 Discrete Cosine Transformation

Following the PCA alignment described above, an AD image is further parameterized using the Discrete cosine transform (DCT). DCT is used for its well-known properties of representing an image in a small number of parameters due to its excellent energy compaction property.

A 2D DCT of an $N \times M$ input image '*f*' is realized through the following equation:

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cos\left(\frac{\pi u}{2M} \cdot (2i+1)\right) \cos\left(\frac{\pi v}{2N} \cdot (2j+1)\right) \quad (4\text{-}5)$$

Where *F(u,v)* is the DCT coefficient at row *u* and column *v* of the DCT matrix. *C(u)* is a normalization factor equal to $1/\sqrt{2}$ for u=0 and 1 otherwise [17].

For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT image. The lower right values represent higher frequencies, and are often small enough to be neglected with little visible distortion. Thus, the ADs image is transformed into the DCT domain and zigzag scanned from the top left corner to select a number of coefficients that represent a smoothed version of the accumulated differences. This process is known as Zonal Coding [18]. The number of scanned coefficients is determined empirically as described in the experimental results section.

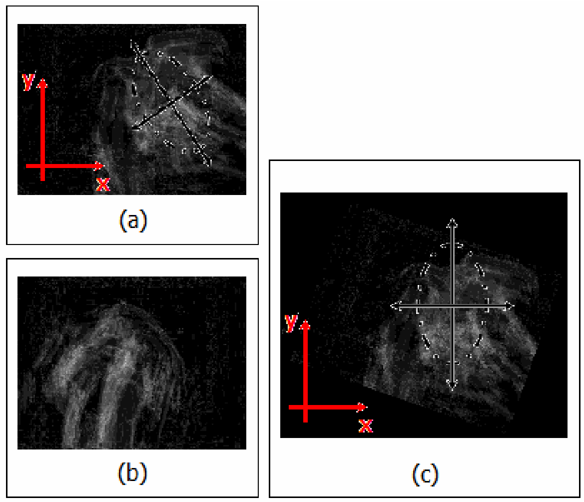

Figure 4-2 (a) Absolute AD for letter ع with tilted handwriting angle, (b) with correct handwriting angle, (c) PCA is applied

4.2.3 ADs Projection

As an alternative approach to 2D DCT transformation, this section proposes the use of image projections for the ADs. The projected data is then transformed into the frequency domain through a 1D DCT transformation. The projection can be applied from any arbitrary angle; this work, however, experiments with vertical and/or horizontal projections. The former captures the horizontal movement while the latter captures the vertical movement of the hand.

4.2.4 Walsh - Hadamard Transform

Another spatial transform is used in this thesis in order to reduce the images' contents into small parameters. This is characterized by low complexity compared to other discrete image transformations like DCT. This transformation is based on a matrix composed of +1s and -1s element. For a given input image $f(i, j)$ of a size of $N \times N$ we can have:

$$F(u,v) = \frac{1}{\sqrt{MN}} \prod_{i=0}^{N-1} \prod_{j=0}^{M-1} f(i, j)(-1^{b(i)*b_{N-1-i}(u)})(-1^{b(j)*b_{N-1-j}(v)}) \quad (4\text{-}6)$$

Where $b(i)$ & $b(j)$ are amplitude level of the input image [16].

After this transformation, we apply zonal coding in order to restore the most important information in the resultant transformation.

4.3 Feature Extraction Schemes

This section proposes various schemes for combining the above-mentioned feature extraction techniques.

4.3.1 Absolute Accumulative Differences

In this approach, motion detection is stored in the absolute accumulative image differences as shown in the figure 4-3. This approach cannot recognize hand movements' directions, for instance, right or left movements are considered as one motion pattern segment.

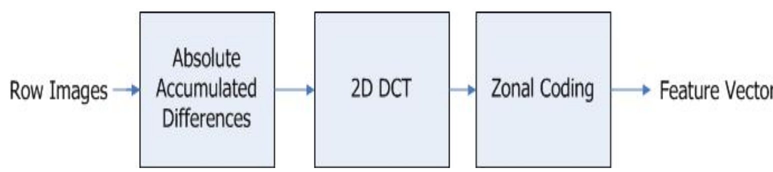

Figure 4-3 Absolute AD

4.3.2 Polar Accumulative Differences

In this approach, the Positive and Negative ADs are concatenated together as an image prior to the 2D DCT transformation as shown in figure 4-4. Zonal coding is applied with different cutoff values. This method preserves the direction of the hand movements.

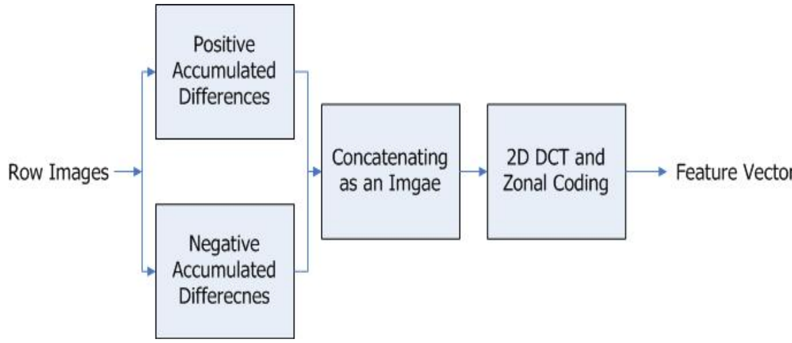

Figure 4-4 Polar ADs

4.3.3 Vectorized Accumulative Differences

Here the positive and negative ADs are 2D DCT transformed followed by zonal coding. The concatenation is then applied to the vectorized DCT coefficients as shown in figure 4-5.

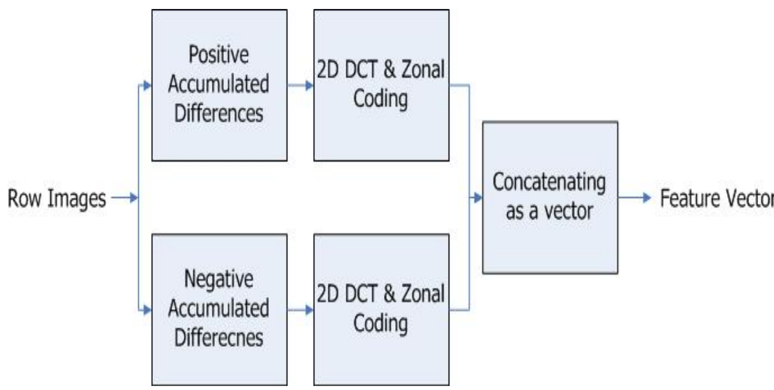

Figure 4-5 Vector ADs

4.3.4 1-D Projected Accumulative Differences

The concatenated image results from Polar ADs can be projected horizontally or vertically and then transformed into the DCT domain. A number of low frequency coefficients can be, then selected to represent the ADs as shown in Figure 4-6.

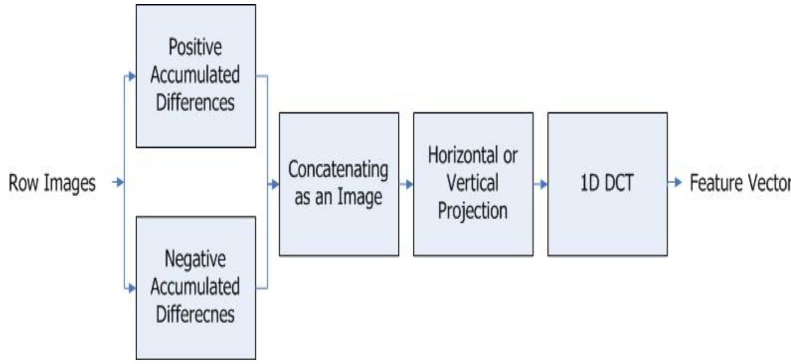

Figure 4-6 1D - Projected Ads

4.3.5 2-D Projected Accumulative Differences

To preserve both horizontal and vertical movements of the hand, both projection schemes are employed. In this case, the low frequency coefficients of both projections are concatenated into one vector. Figure 4-7 shows a block diagram of the proposed solution in conjunction with the Polar ADs scheme.

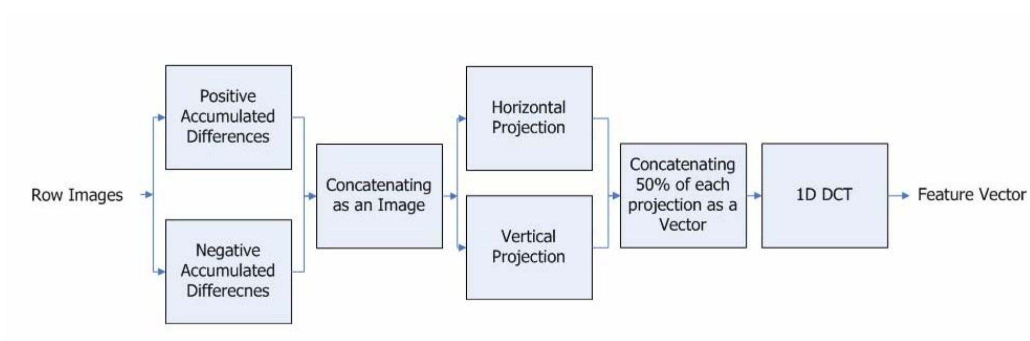

Figure 4-7 2D – Projected ADs

4.3.6 Two-tier weighted Accumulative Differences:

This section proposes to cluster commonly misclassified data into super classes. Super classes are empirically determined, for instance, one super class can contain (ع and غ), (ن and ث) or (ش and ض). If a test data falls into any such super class, a second tier of feature extractions is required. Namely, weighted ADs is applied to the test data followed by any of the spatial feature extraction schemes mentioned above. The new features are then fed to the classifier as illustrated in Figure 4-8.

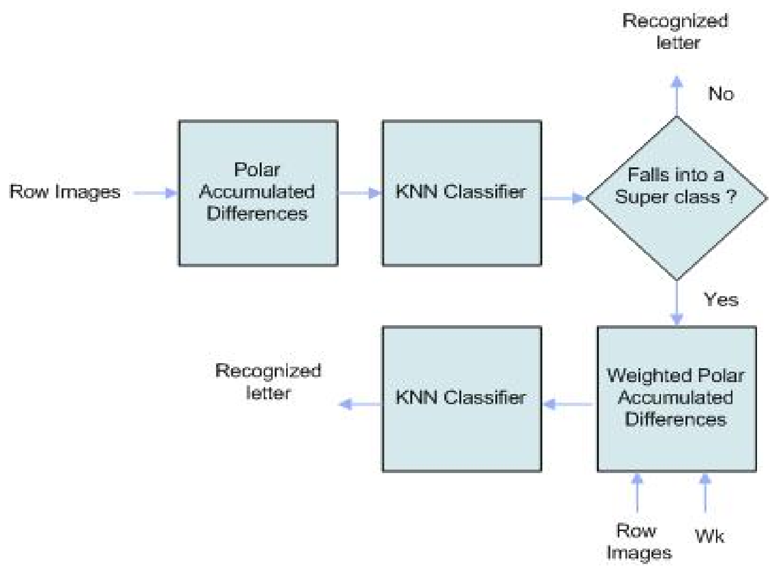

Figure 4-8 Two-tier weighted Accumulative Differences

4.3.7 Walsh – Hadamard (WH):

In this scheme, we threshold the image differences of row images and then apply WH Transform followed by zonal coding as shown in figure 4-9. The resultant feature vector is used in classification using Hidden Markov Models as will be described in the following section.

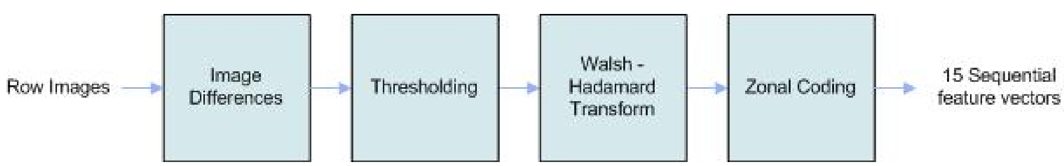

Figure 4-9 WH Transform Scheme

CHAPTER 5

CLASSIFICATION

We introduce two classification methods in this thesis. A classifier based on K Nearest neighbor algorithm which doesn't depend on temporal information and the other one is based on Hidden Markov Model that needs to preserve temporal information for classification.

5.1 KNN Classifiers.

K Nearest neighbor algorithm uses neighborhood samples as predicted values of the test data. It works based on minimum distance from the test set to the training samples to determine the K-nearest neighbors [15]. In this work the following distance measure is used:

$$dist_j = \frac{\mathrm{L_p}(\mathbf{v}_{\mathrm{test}} - \mathbf{v}_{\mathrm{train_j}})}{\mathrm{L_p}(\mathbf{v}_{\mathrm{test}} + \mathbf{v}_{\mathrm{train_j}})} \qquad (5\text{-}1)$$

Where $\mathbf{v}_{test}$ is the output of the feature extraction process as previously outlined for a given test video sequence. Similarly, $\mathbf{v}_{\mathrm{train_j}}$ is the output of the feature extraction process for the j^{th} training video sequence. Finally, L_p is the Minkowski metric defined as

$$L_p(\mathbf{x}) = \left(\sum_i |x_i|^p \right)^{1/p} \qquad (5\text{-}2)$$

Typical values for p are 1 and 2 corresponding to the Manhattan and Euclidean metrics respectively.

5.1.1 Leave N out estimation

These estimations depends on the number of samples left and considered as testing sets, and the remaining samples are considered as training sets. As result, we have a Leave One Out scenario where 1 sample/class is used for testing and we have a Leave Two Out scenario where 2 samples/class are used for testing. In this thesis, the second one is used extensively for its reliability. Another factor in these methods is the number of minimum distances K per class i.e. specifying the majority vote for a distance between the tested value and the trained one to be considered as minimum distance. In this thesis we used range of K values which increases the probability of a minimum distance to match the letter class [20].

5.1.2 Classification Evaluation

Confusion matrix (CM) is a technique that is used to evaluate the performance of a classifier which stores the correct predicted values diagonally on a class by class basis in the optimum classification cases and stores the incorrect ones elsewhere in the matrix in the misclassification cases [14].

In our thesis, CM is composed of 28x28 elements corresponding to the total number of Arabic letters and each letter's class has an optimal number of correct classified values stored diagonally which is calculated using the following formula:

$$\left(\frac{\textit{Total Number of Samples}}{\textit{Number of Samples left out}}\right)=\left(\frac{\textit{Total Number of Samples}!}{\textit{Number of Training samples}!}\right) \qquad (5\text{-}3)$$

where ! is the evaluation of $n! = n \times (n-1) \times (n-2) \times \ldots$

In case of Leave One Out, the typical number of correct values stored diagonally per letter's class is 8 and in case of Leave Two Out we have 56. For this, the typical confusion matrix for Leave Two Out case is formed as follows:

$$
\begin{array}{cc}
 & \begin{array}{cccccc} \text{أ} & \text{ب} & \text{ت} & \cdots & \text{و} & \text{ي} \end{array} \\
\begin{array}{c} \text{أ} \\ \text{ب} \\ \text{ت} \\ \vdots \\ \text{و} \\ \text{ي} \end{array} &
\begin{pmatrix}
56 & 0 & 0 & 0 & 0 & 0 \\
0 & 56 & 0 & 0 & 0 & 0 \\
0 & 0 & 56 & 0 & 0 & 0 \\
0 & 0 & 0 & \ddots & 0 & 0 \\
0 & 0 & 0 & 0 & 56 & 0 \\
0 & 0 & 0 & 0 & 0 & 56
\end{pmatrix}
\end{array}
$$

Figure 5-1 Leave Two Out confusion matrix

There are number of useful measures used to show the performance of the CM and so the classifier recognition performance as shown below:

$$\text{Recognition Rate / Classifier} = \frac{\sum Diagonal\,Values\,(Correct)}{\sum All\,Values\,(Correct\,\&\,Incorrect)} \quad (5\text{-}4)$$

$$\text{Correct Rate / Class} = \frac{\sum Correct\ of\ that\ Class}{\sum Correct\,\&\,Incorrect\ of\ that\ Class} \quad (5\text{-}5)$$

$$\text{Error Rate / Class} = \frac{\sum Incorrect\ of\ that\ Class}{\sum Correct\,\&\,Incorrect\ of\ that\ Class} \quad (5\text{-}6)$$

5.2 HMM Classifier.

5.2.1 Letter modeling using HMM

Hidden Markov Model is stochastic modeling methodology, which is composed of hidden data sequences with relevant temporal information and observable ones with relevant spatial information. For these characteristics, HMM is a perfect technique used to describe hand movements dynamics.

Any Hidden Markov Model can be described analytically as follows[13]:

5.2.1.1 Hidden States Probability A:

For a given Hidden states vector $S = (s_1, s_{2,}\ldots, s_N)$, there is $N \times N$ state transition probability matrix A, where each element represents the probability between the current and previous states as follows:

$$A = \{a_{ij}\} = \{P(s_t = j \mid s_{t-1} = i)\} \quad :1 \le i, j \le N \quad (5\text{-}7)$$

In our system, the changing direction of the hand movement during writing a letter is considered as hidden state. Figure 5-2 below shows the number of states in letter (ب). Each letter has a distinct number of hidden states recognized whenever the letter pattern direction is changed. Moreover, the movements used to set the letter dot is considered as a one state as shown.

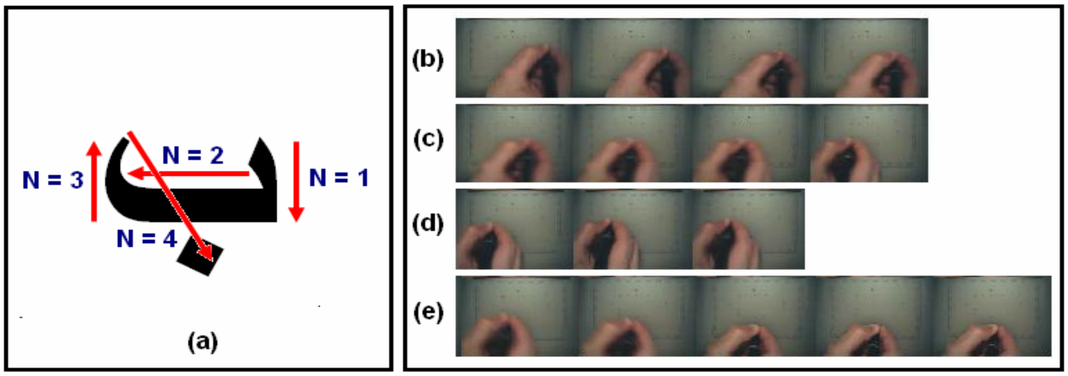

Figure 5-2 Hidden states for writing letter ب. In (b), **4** frames describe hand movements for N = 1, in (c) 4 frames for N = 4, and in (d) three frames for N = 3 and 5 frames for N = 5 represents the dot movements.

These states are related to each other in a hidden Markov chain as shown in figure 5-3. Each state is only related to the previous one where the nodes are the states and the transition probabilities are the links.

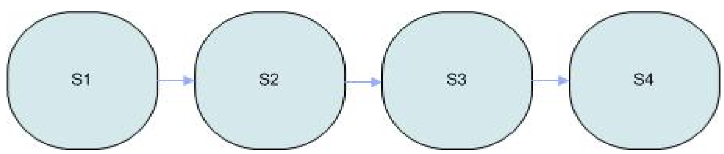

Figure 5-3 Hidden Markov Chains for letter ب

5.2.1.2 Observable States Probability B :

For given observable states $O=(o_1, o_2, \cdots, o_M)$, there is $N \times M$ probability matrix B that relates the hidden states with the observable ones, where each element is formulated as follows:

$$B=\{b_i(M)\}=\{P(O_t=o \mid S_t=i)\} \quad :1 \le i \le N, 1 \le o \le M \qquad (5\text{-}7)$$

In our system, the difference between two consecutive frames is considered as observable state O .

5.2.1.3 Initial states probability π :

Another probability value is used to construct the HMM which is a vector that describe the initial state of the model at $t=0$, it is as follows:

$$\pi = \{\pi\} = \{P(s_1 = i)\} \quad : 1 \le i \le N \quad (5\text{-}8)$$

5.2.2 The learning process:

Generate an HMM model that adequately characterizes each letter using Baum – Welsh algorithm. It's Expectation – Maximization that determines the maximum likelihood among the observable events.

For a given letter class G , there is HMM model that characterizes that class as follows:

$$\lambda_G(\pi, A, B) \quad (5\text{-}9)$$

Where λ_G is adjusted in order to maximize the probability of the total likelihood of the observations is given as follows:

$$P(O \mid \lambda_G) = \sum_S P(O \mid S, \lambda_G) P(S \mid \lambda_G) \quad (5\text{-}10)$$

$$= \pi_1 b_1(o_1) a_{11} b_1(o_2) a_{12} b_2(o_3) a_{13} b_3(o_3) \ldots$$

This maximization is achieved iteratively using Baum – Welsh algorithm as shown in figure 3-4.

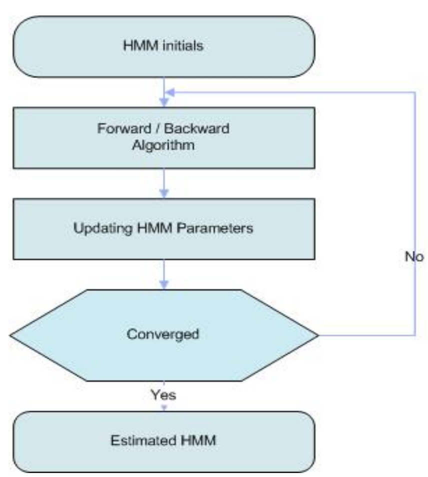

Figure 5-4 Baum – Welsh Algorithm

The structure of the algorithm has the following processing steps:

5.2.2.1 Forward Processing:

Assume that $\alpha_t(i) = P(o_1, o_2, \cdots, o_t, s_t = i \mid \lambda_G)$, then to compute the likelihoods $\alpha_t(i)$ we use the following steps:

- Initialization: $\alpha_1(i) = \pi_i b_i(o_1) \quad : 1 \le i \le N$ (5-11)
- Induction: For $t = 2,3,4,\cdots,T$,

$$\alpha_t(j) = \left(\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right) b_j(o_t) \quad : 1 \le j \le N \quad (5\text{-}12)$$

- Termination: $P(O \mid \lambda_G) = \sum_{i=1}^{N} \alpha_T(i)$ (5-13)

5.2.2.2 Backward Processing:

Assume that $\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid s_t = i, \lambda_G)$, then to compute the likelihoods $\beta_t(i)$ we follow these steps:

- Initialization: $\beta_T(i) = 1 \quad : 1 \le i \le N$ (5-14)
- Induction: For $t = T-1, T-2, \ldots, 1$,

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad : 1 \le i \le N \quad (5\text{-}15)$$

- Termination: $P(O \mid \lambda_G) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i)$ (5-16)

5.2.2.3 Parallel Processing:

We estimate number of parameters used in modeling HMM as follows:

- Occupation likelihoods: $\gamma_t(j) = \dfrac{\alpha_t(j)\beta_t(j)}{P(O \mid \lambda_G)}$ (5-17)
- Transition likelihoods: $\zeta_t(i,j) = \dfrac{\alpha_{t-1}(i) a_{ij} b_j(o_t) \beta_t(j)}{P(O \mid \lambda_G)}$ (5-18)
- Transition accumulators:

$$\underline{\pi_i} = \underline{\pi_i} + \gamma_1(i) \quad (5\text{-}19)$$

$$\underline{a_{ij}} = \dot{\underline{a_{ij}}} + \sum_{t=2}^{T} \zeta_t(i, j) \quad (5\text{-}20)$$

$$\overline{a_i} = \dot{\overline{a_i}} + \sum_{t=2}^{T} \gamma_t(i) \quad (5\text{-}21)$$

Where $\dot{s}$ denotes to the previous value of the accumulator s.

- Output accumulators:

$$\underline{\mu_i} = \dot{\underline{\mu_i}} + \sum_{t=1}^{T} \gamma_t(i) o_t \quad (5\text{-}22)$$

$$\underline{\Sigma_i} = \dot{\underline{\Sigma_i}} + \sum_{t=1}^{T} \gamma_t(i)(o_t - \mu_i)(o_t - \mu_i)' \quad (5\text{-}23)$$

$$\overline{b_i} = \dot{\overline{b_i}} + \sum_{t=1}^{T} \gamma_t(i) \quad (5\text{-}24)$$

- Repetition processing:

For all the results in the training sets:

- Recompute the forward and backward likelihoods.
- Recompute the occupation and transition likelihoods.
- Increment the accumulators.
- Update processing:

Finally we update the models as follows:

$$\hat{\pi}_i = \frac{1}{R} \sum_{r=1}^{R} \underline{\pi}_i \quad (5\text{-}25)$$

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \underline{a_{ij}}}{\sum_{r=1}^{R} \overline{a}_i} \quad (5\text{-}26)$$

$$\hat{b}_i \begin{cases} \hat{\mu}_i = \dfrac{\sum_{r=1}^{R} \underline{\mu}_i}{\sum_{r=1}^{R} \overline{b}_i} \\ \hat{\Sigma}_i = \dfrac{\sum_{r=1}^{R} \underline{\Sigma}_i}{\sum_{r=1}^{R} \overline{b}_i} \end{cases} \quad (5\text{-}27)$$

5.2.3 The Validating process:

The algorithm used is forward one which is described as follows:

Assume that $\alpha_t(i) = P(o_1, o_2, \cdots, o_t, s_t = i \mid \lambda_G)$, then to compute the likelihoods $\alpha_t(i)$ we use the following steps:

- Initialization: $\alpha_1(i) = \pi_i b_i(o_1) \quad : 1 \le i \le N$ (5-28)
- Induction: For $t = 2,3,4,\cdots,T$,

$$\alpha_t(j) = \left(\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right) b_j(o_t) \quad : 1 \le j \le N \qquad (5\text{-}29)$$

- Termination: $P(O \mid \lambda_G) = \sum_{i=1}^{N} \alpha_T(i)$ (5-30)

5.2.4 The evaluation method:

Data sets are increased to 40 samples per letter, where 30 samples are used for training and 10 ones are used for testing. Number of hidden states is manually set into the program. The hidden states range from one state for letter (أ) to 9 states as in letter (ش). The zonal cutoff values used ranges from 10 to 100. Figure 5-5 shows the processing stages required to evaluate the HMM classifier.

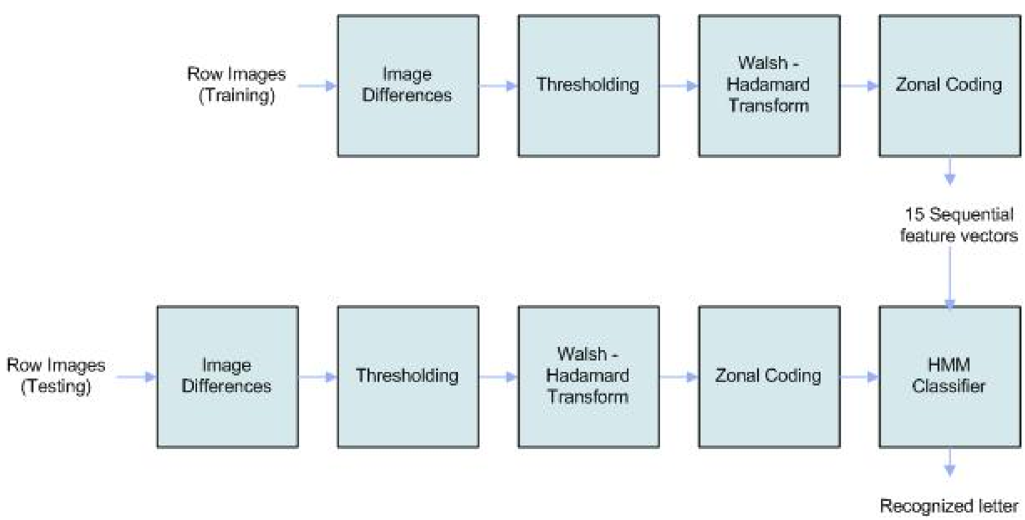

Figure 5-5 HMM Recognizer

CHAPTER 6

EXPERIMENTAL RESULTS AND CONCLUSION

6.1 KNN Classification

The zonal cutoff of the 2D DCT coefficients is determined empirically in Figure 6-1. The figure implements the KNN classifier and experiments with 3 different values for ‘K’, the majority vote . It is shown that the highest recognition rate is realized at a zonal cutoff of 90 DCT coefficients. In other words, the whole image sequence of a given letter can be represented by as few as 90 DCT coefficients with a rather high recognition rate. The figure also shows that setting ‘K’ to 5 produces higher recognition rates.

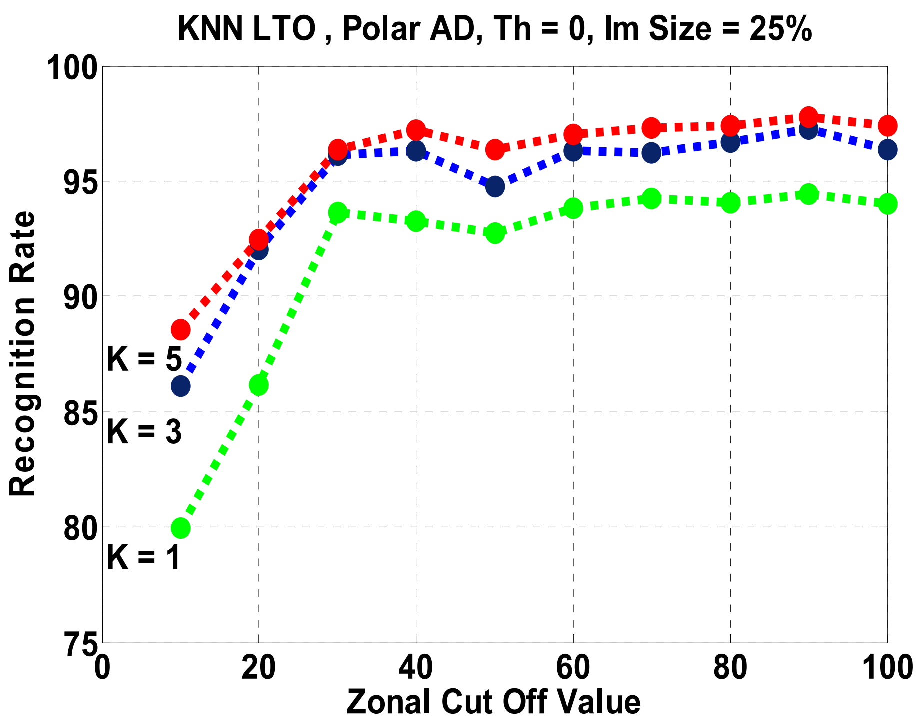

Figure 6-1 Zonal Cutoffs and classifier's recognition rate

The computational load can be further reduced by considering the dimensions of the individual row images that represent a given letter. As a result, temporal and spatial analyses are applied to a set of images with reduced dimensions. Figure 6-2 shows that reducing the input size to 25% is feasible.

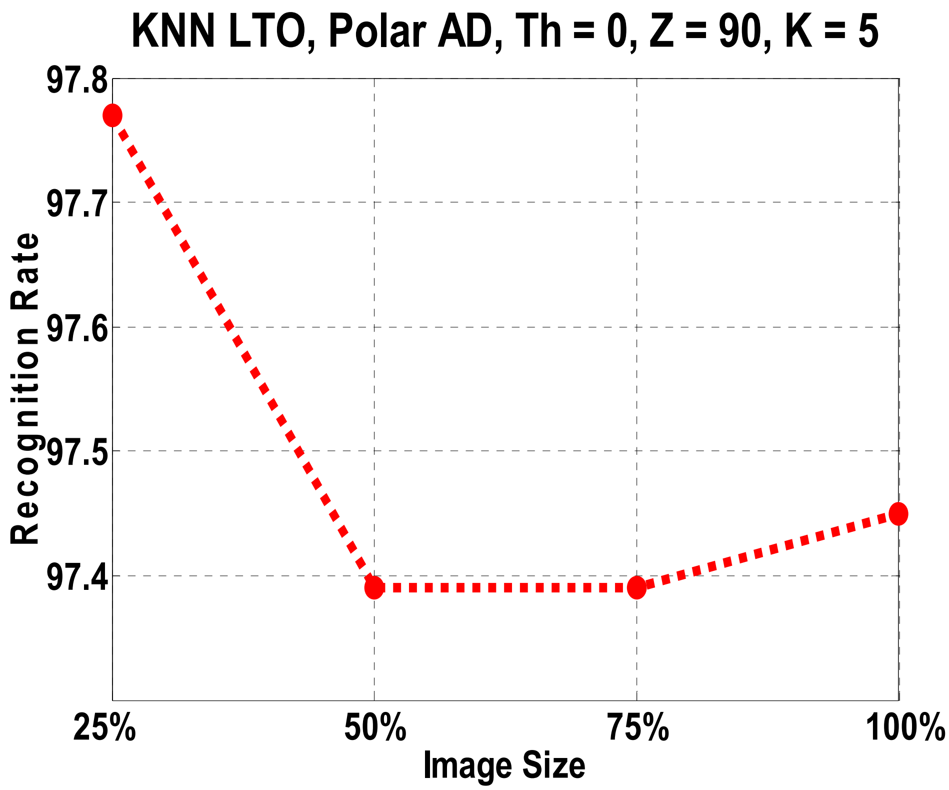

Figure 6-2 Image Size and recognition rate

As mentioned in Section 4, the Threshold of the ADs is set to either 0, mean or pixel difference exceeding the threshold value. Figure 6-3 shows that the highest recognition rate is realized through setting the Threshold to 0 followed by mean and greater than the mean.

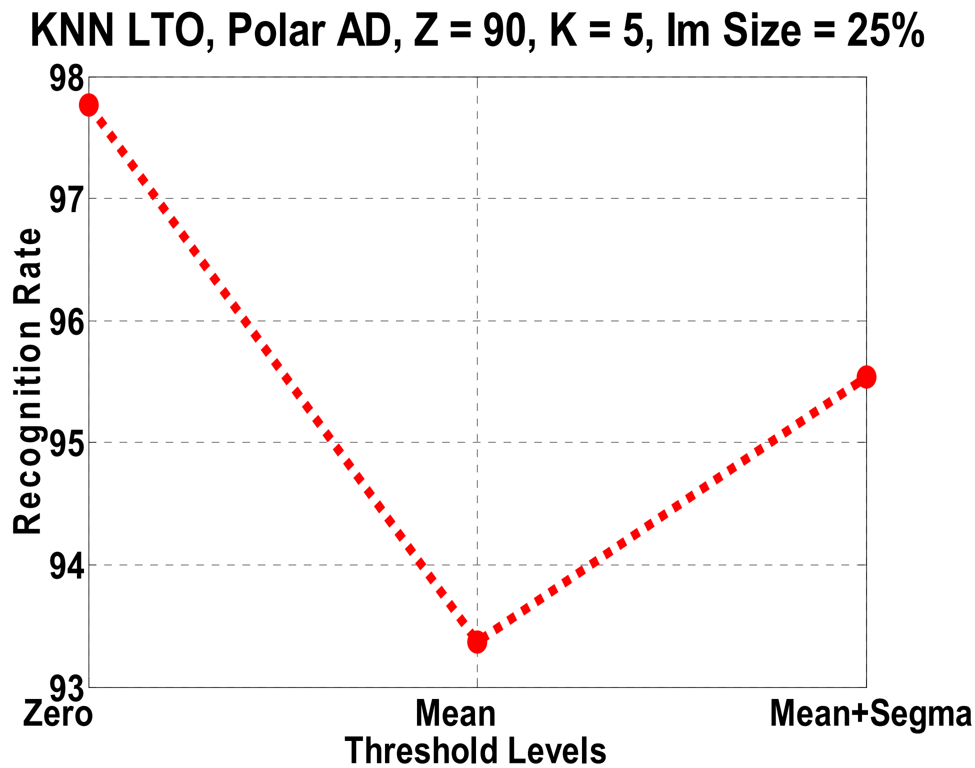

Figure 6-3 Threshold and recognition rate

Having determined the optimal parameters of the temporal and spatial feature extractions, Table 6-1 compares between the recognition rates of various ADs schemes for two different users and shows the average time required for the feature extraction process and the classification one for a single letter.

Table 6-1 Recognition rate of different ADs schemes for two different users

Accumulative Difference Scheme	User 1 (%)	User 2 (%)	$T_{feature}$	$T_{classification}$
Absolute AD	87.76	85.33	4.76 m	0.0329 s
Polar AD	97.77	96.86	53.14 s	0.0296 s
Vectorized AD	93.20	92.80	52.20 s	0.0417 s
Horizontal Projection AD	57.78	56.78	54.54 s	0.0435 s
Vertical Projection AD	86.86	84.92	55.95 s	0.0443 s
2D Projection AD	87.95	86.20	1.51 m	0.0482 s
Two-tier AD	99.11	98.60	53.74 s	0.0285 s

Clearly since the polar approach preserves the direction of motion in the absolute ADs, its recognition rate was superior to the absolute ADs. Also, the concatenation of the vertical and horizontal projections of the ADs outperformed the 1D projection which results in losing the location of dots in letters as pointed out earlier. This problem is pronounced in the 'horizontal only' projection since the vertical location of the dot(s) is vital. We can also notice how close and successful are the handwriting recognition rates for the two different users, revealing the quality of the accumulative image differences schemes which are applied, especially Polar ADs and Two-tier one.

Finally, the confusion matrices are examined for the best two cases of the proposed solutions in Table 2. Throughout all the experiments, LTO strategy is used to partition the data into training and test data, and to increase the statistical significance of the recognition rates. Applying LTO to our database which is composed of 8 repetitions per letter, generates 56 test patterns per letter. Figure 6-4 shows the confusion matrix of the recognition results using the following feature extraction and classification methodology. First, temporal analysis is applied to the frames of a video sequence by down-sampling them to ¼ their original size followed by computing their polar ADs using 0 threshold. Second, spatial analysis is done by computing 90 zonal DCT parameters to represent the feature vector of a given video sequence. Classification is done using a KNN classifier with K=5. Figure 6-4 shows that 35 out of 1568 patterns were misrecognized for user 1 corresponding to a recognition rate of 97.77%. It also shows that the significant mis-recognitions (greater than 5%) are mainly concentrated in five letters. In the other hand, user 2 shows some close recognition rates like 96.86% but totally different errors locations, which is naturally due to human writing variations. Figure 6-5 shows that 49 errors out of 1568 patterns were misrecognized using KNN classifier with similar setting that was applied for user 1.

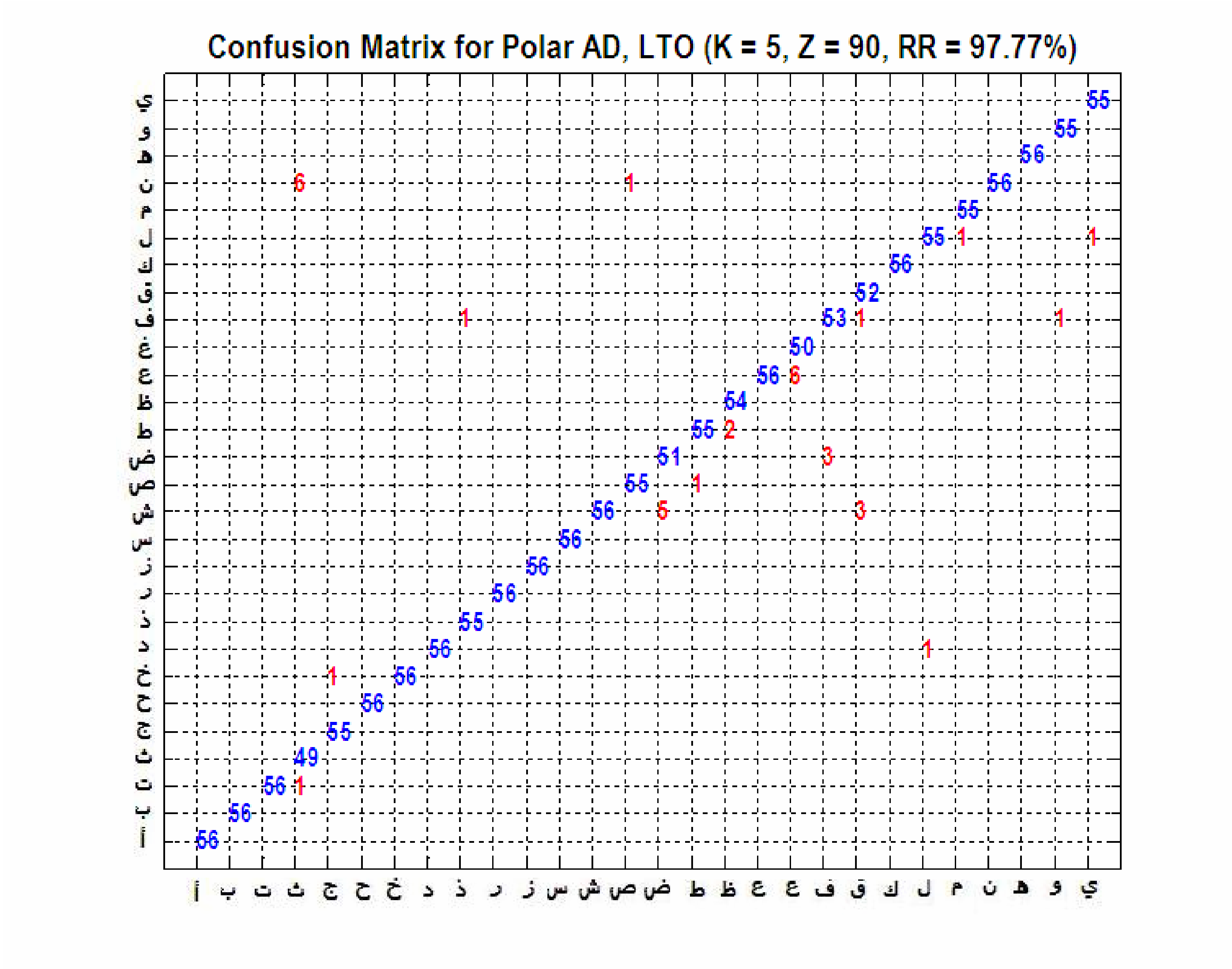

Figure 6-4 Confusion Matrix for the Polar ADs for user 1

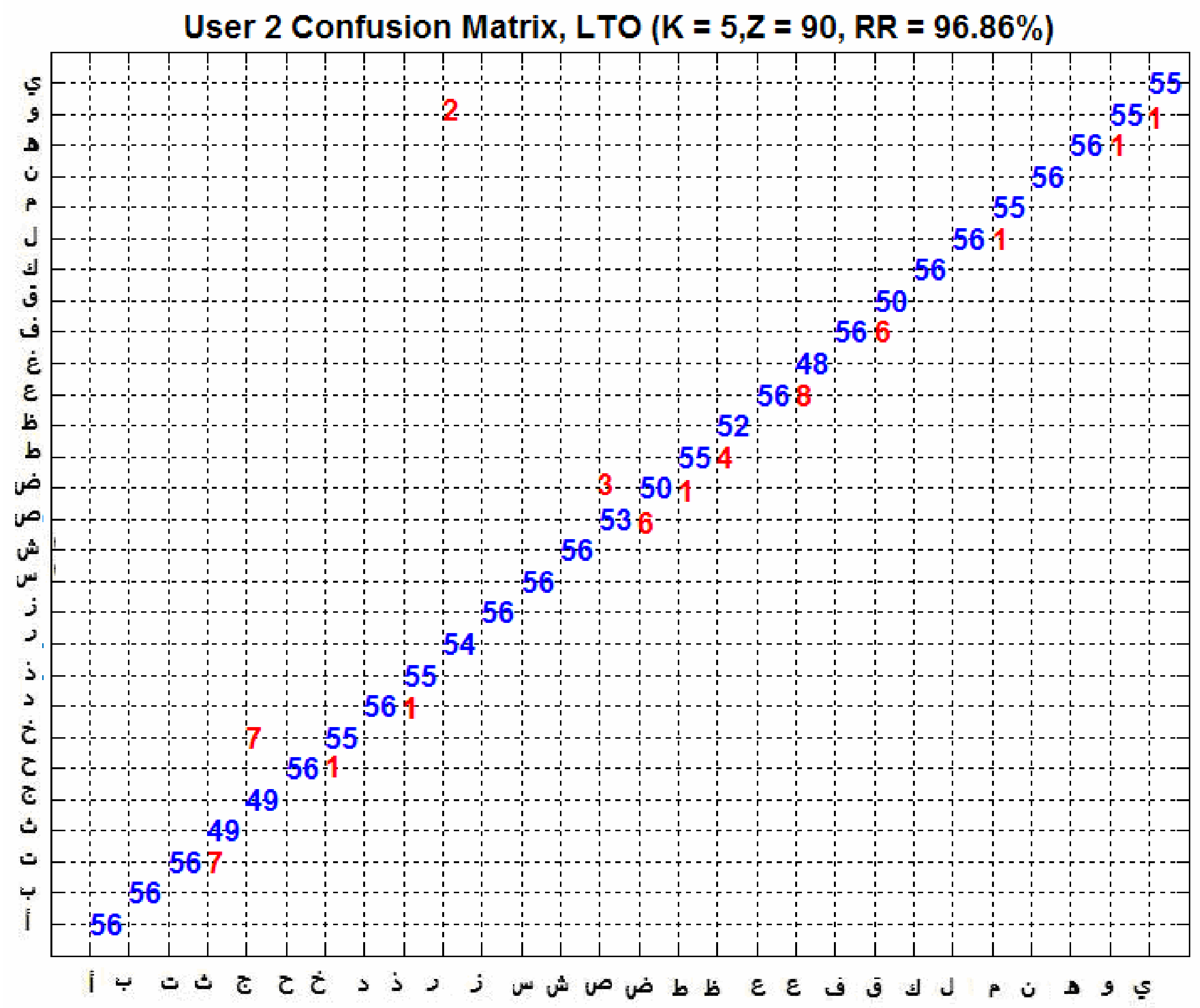

Figure 6-5 Confusion Matrix for the Polar ADs for user 2

On the other hand, a two-tier weighted AD scheme improves the recognition rate to 99.11% for user 1 where super classification is applied for letters (ن ث) (ش ض) and (غ ع) as shown in Figure 6-6. For user 2, the recognition rate is increased into 98.60% and super classification is applied for letters (ث ت) , (خ ج), (ص ض), (ظ ط) (غ ع) and (ق ف) as shown in Figure 6-7 These enhancements are done by correcting the significant errors committed using polar ADs.

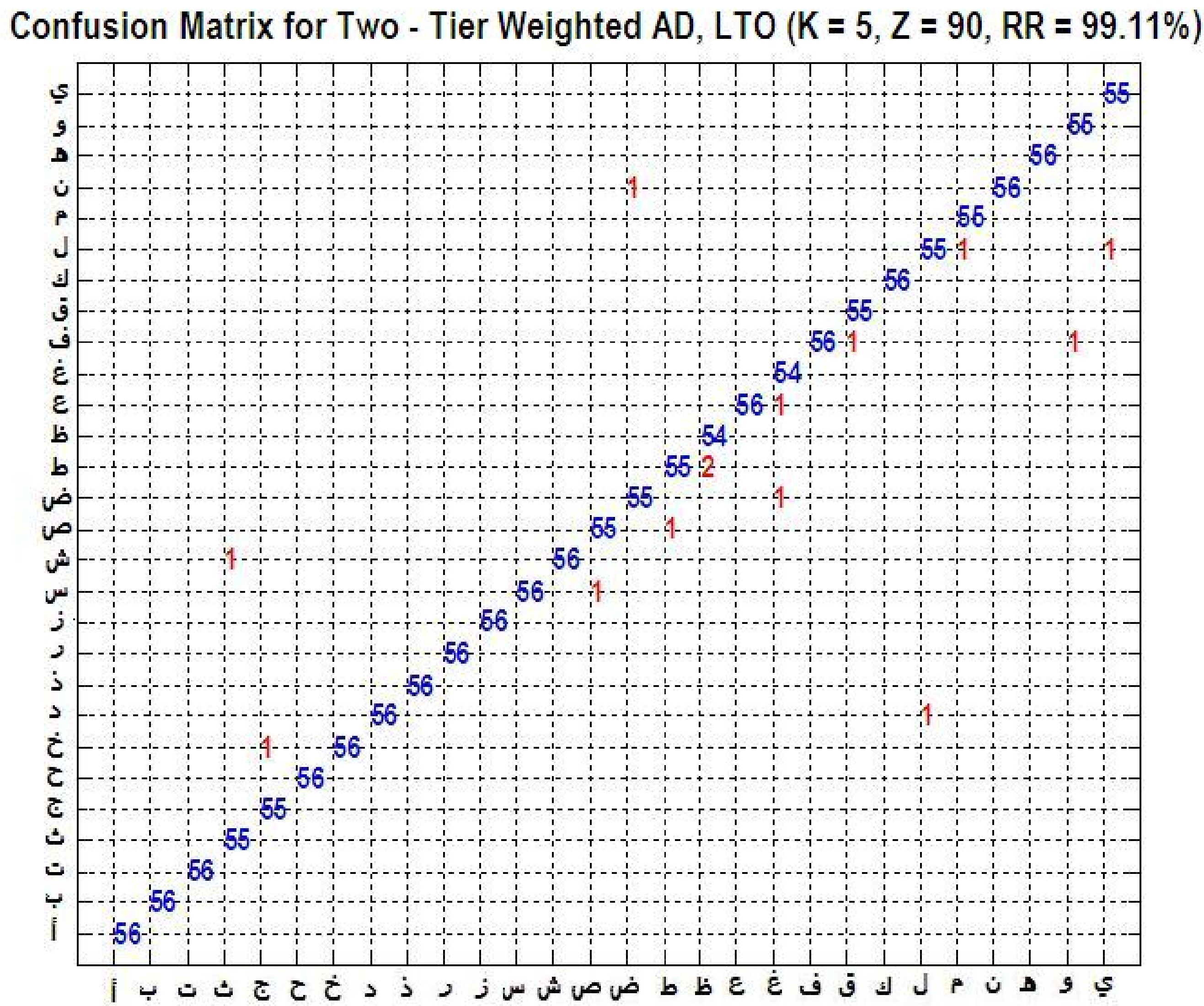

Figure 6-6 Performance of two-tier weighted ADs for user 1

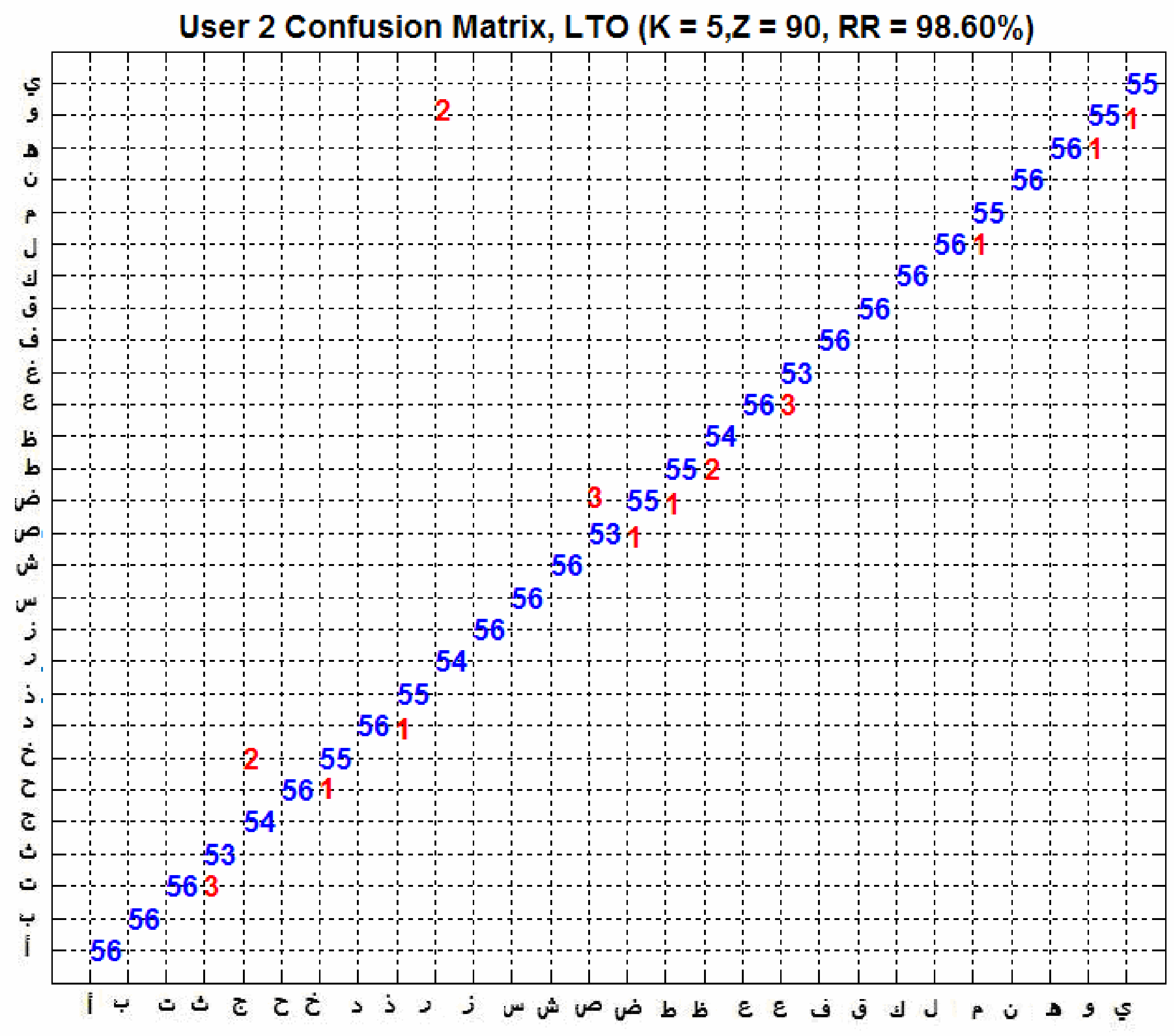

Figure 6-7 Performance of two-tier weighted ADs for user 2

6.2 HMM Classification

The left to right HMM architecture is used in the following experiment. A state can only transit to its immediate right neighbor or stay in the same state. The training method applied is the Baum-Welch algorithm. The number of states is empirically determined to be 2-4 according to the complexity of the letter.

In this experiment the temporal information of the input image sequence is preserved. The feature extraction processing preserves the absolute differences between successive images without accumulating them into one image.

The absolute image differences are then threshold to a value that preserves the main contents of the images. The resultant data is then transformed using Walsh – Hadamard transform to reduce the complexity of the data and ease the next processing steps which are used to model the HMM. Zonal coding is also applied to reduce the complexity of the images by concentrating on low frequency contents.

The results of using HMM recognizer are promising ones, where recognition is maximized at cutoff value of 20, 30 & 50 to 96.43% for the first user and maximized at cutoff value of 50 to 91.11% for the second user as shown in figure 6-8.

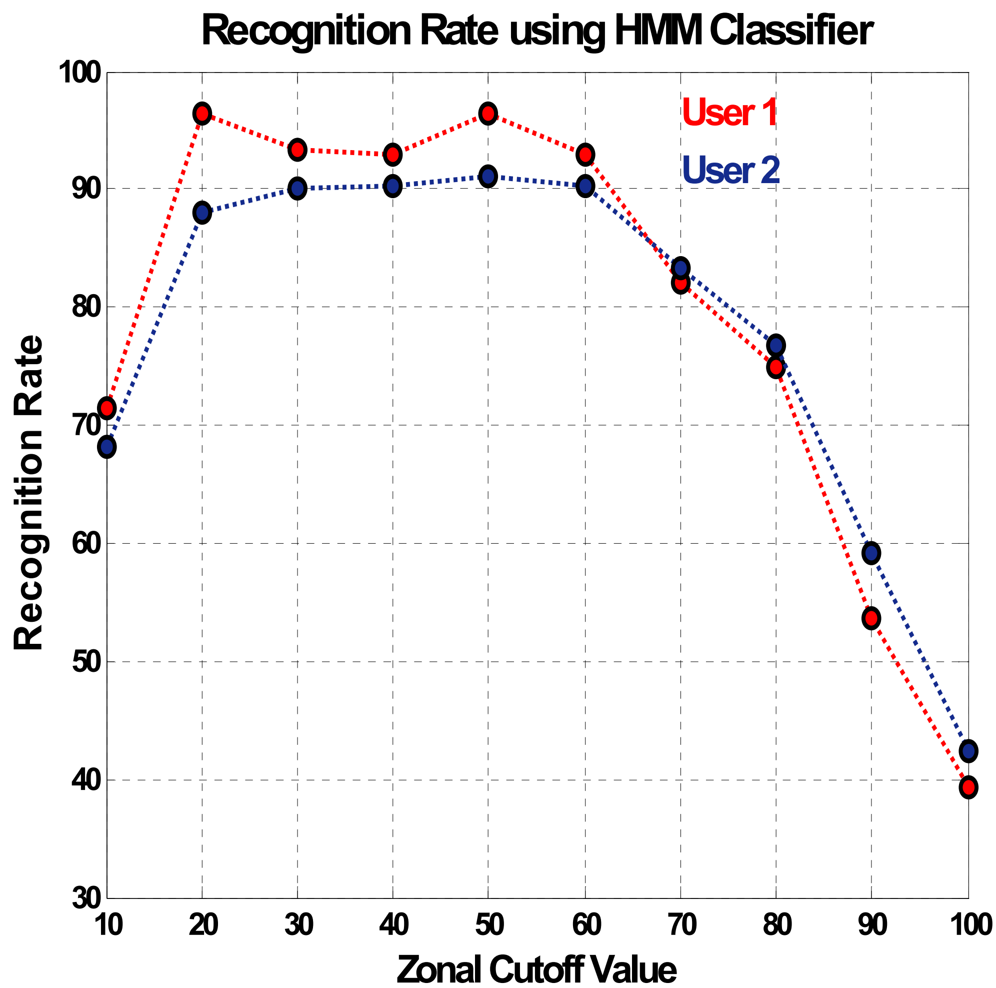

Figure 6-8 Results of using HMM recognizer

Figure 6-8 also shows that HMM classifier is showing lower recognition rates compared to KNN one by 2.68 % for user 1 and 8.81% for user 2. This is can be clarified because HMM is a statistical modeling method which requires large amount of datasets in the training phase in order to have high recognition rates, while KNN produces high results with the same amount of information without mentioning the simplicity of KNN classifier.

6.3 Conclusion

In this paper, we presented a novel solution for the online video-based Arabic handwriting recognition problem. A variation of temporal and spatial features was examined. Different schemes of accumulated differences were used to compress the motion information in the video sequence of the writer's hand into a static image. The resulting AD image is further parameterized using DCT and line projections. PCA is employed to align the accumulated differences rendering the feature extraction process robust to rotation. A KNN classifier is used and yielded excellent recognition rates as high as 97.77% with Polar ADs and 99.11% for the two-tier weighted AD scheme. The system was also checked with another user that shows close successful recognition rates upto 96.86% with Polar ADs 98.60% for two-tier AD scheme with totally different error locations due to variation in human handwritten letters.

In the other hand, using HMM recognition system shows 96.43% for user 1 and 91.11 % for user 2. The results in comparing them with KNN recognition revealed how the obtained figures are close. This increases the quality of feature extraction schemes that are being developed in this thesis.

The experiments show that, KNN classification has a bit higher recognition rate as compared to HMM one. This is because of preserving temporal information in case of HMM.

This area of research still need much work to be done like enhancing the time required for extracting the features. Also, experimenting these schemes with non-isolated letters.

REFERENCES

[1] Jong Oh, "An On-Line Handwriting Recognizer with Fisher Matching, Hypotheses Propagation Network and Context", Ph.D. dissertation, New York University, Department of Computer Science, May 2001.

[2] Al-Emami, S. and Usher, M. “On-Line Recognition of Handwritten Arabic Characters”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 7, July 1990. pp. 704-710.

[3] Bouslama, F. and Amin, A., “Pen-based Recognition System of Arabic Character Utilizing Structural and Fuzzy Techniques”, 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia. Editors, L.C. Jain and R.K. Jain, pp 76-85.

[4] Alimi, A. and Ghorbel, O. “The Analysis of Error in an On-Line Recognition System of Arabic Handwritten Characters”, Proceedings of ICDAR 1995, 14-16 August 1995,Montreal, Canada. pp. 890-893.

[5] El-Wakil,M. and Shoukry, A., “On-Line Recognition of Handwritten Isolated Arabic Characters”, Pattern Recognition, vol. 22, no.2, pp. 97-105, February 1989.

[6] Alimi, A., “An Evolutionary Neuro-Fuzzy Approach to Recognize On-Line Arabic Handwriting”, Proceedings of the 4th International Conference Document Analysis and Recognition (ICDAR '97), pp. 382-386, 1997.

[7] Neila Mezghani, Mohamed Cheriet and Amar Mitiche. , " Combination of Pruned Kohonen Maps for On-line Arabic Characters Recognition ", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), v 2, p. 900, 2003.

[8] Tim Klassen, “Towards Neural Network Recognition Of Handwritten Arabic Letters”, Master Thesis of Computer Science, Dalhousie University Halifax, Nova Scotia, 2001.

[9] O¨ zcan O¨ksu¨z, Ugˇur Gu¨du¨kbay , “Video Based On line Handwriting Recognition”, Dept. of Computer Eng. Bilkent University.
[10] Feng Lin, Xiaoou Tang. "Dynamic Stroke Information Analysis for Video-Based Handwritten Chinese Character Recognition," iccv, vol. 01, no. 1, p. 695, Ninth 2003 .

[11] O¨zcan O¨ksu¨z, Ugˇur Gu¨du¨kbay, “ COMPUTER VISION BASED TEXT AND EQUATION EDITOR FOR LATEX ”, in Proc. of IEEE International Conference on Multimedia and Expo (ICME'2004), Vol. 2, pp. 1255-1258, Taipei, Taiwan 2004.

[12] Markus Wienecke, Gernot A. Fink and Gerhard Sagerer, “Towards Automatic Video-based Whiteboard Reading”,. In Proc. Int. Conf. on Document Analysis and Recognition, vol.1, pp 87-91, Edinburgh, Scotland, 3-6 Aug 2003. IEEE .

[13] Philip Jackson , *Course Notes in Vision Speech & Signal Processing* , University of Surrey, Guildford, spring semester, 2004.
[14] Howard Hamilton, *Course Notes in Knowledge Discovery in Databases,* University of Regina, winter semester, 2002.
[15] Jerry Zhu, *Course Notes in Introduction to Artificial Intelligence* , University of Wisconsin - Madison, fall semester, 2005.
[16] Jeremy Johnson, *Course Notes High Performance Computing*, Drexel University, spring semester, 2004.
[17] Syed Ali Khayam, *Seminar in Discrete Cosine Transform theory and applications.* Michigan State University, March, 2003.
[18] William J. Geckle, *Course notes in Image Compression and Packet Video.* Johns Hopkins University, Fall, 2000.
[19] Andrew Ng, *Course notes in Machine Learning.* Stanford University, Autumn 2006.
[20] Andrew Moore, *Course notes in Machine Learning.* Carnegie Mellon University, fall 2005.
[21] El-Sheik, T.S. and El-Taweel, S.G., “Real-Time Arabic Handwritten Character Recognition”, Pattern Recognition, volume 23 (1990), number 12, pp. 1323-1332 .

Appendix

In this section, I included few samples from both users as shown below:

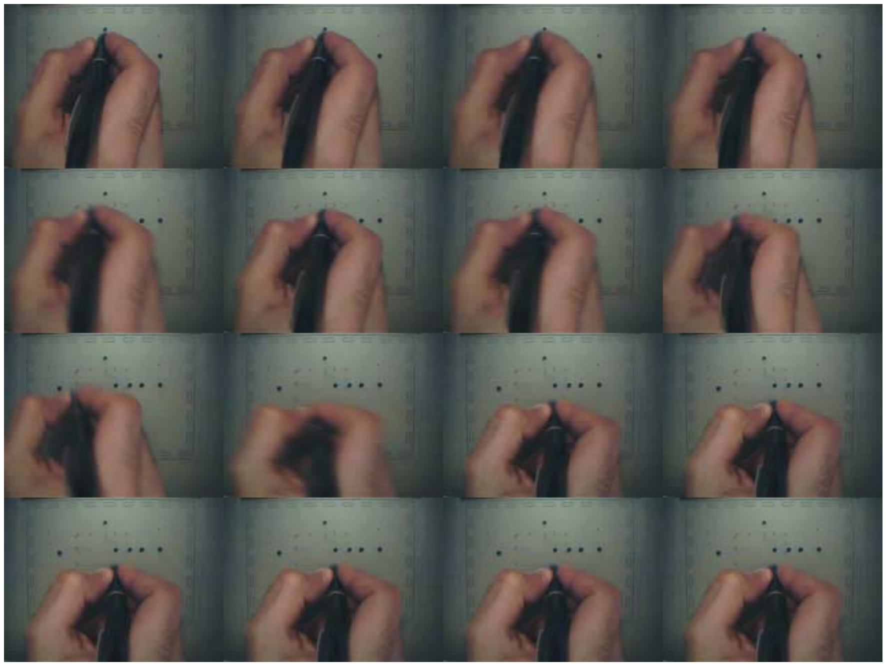

Figure A-1 Example of writing letter (ع) for user 1.

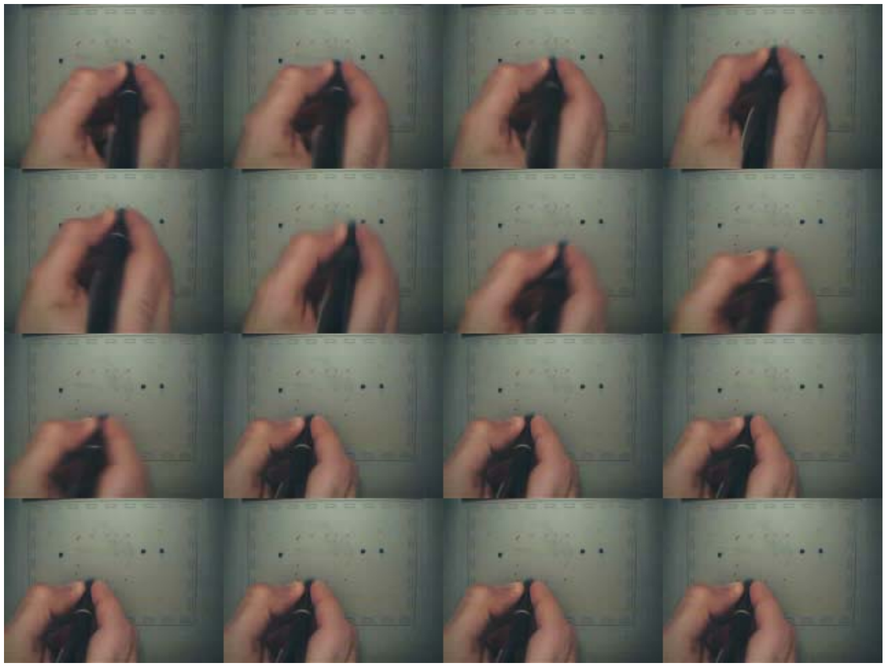

Figure A-**2** Example of writing letter (و) for user 1.

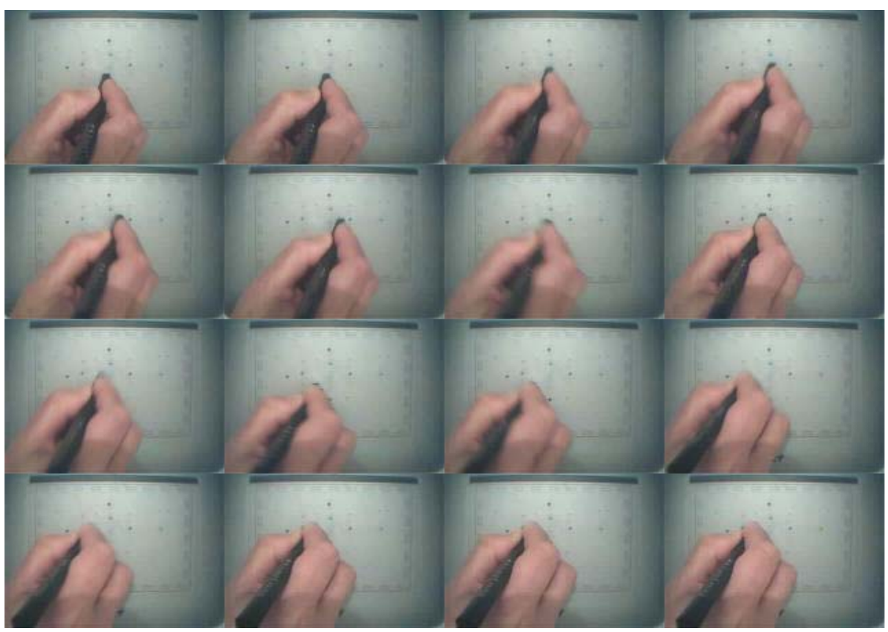

Figure A-3 Example of writing letter (ص) for user 2.

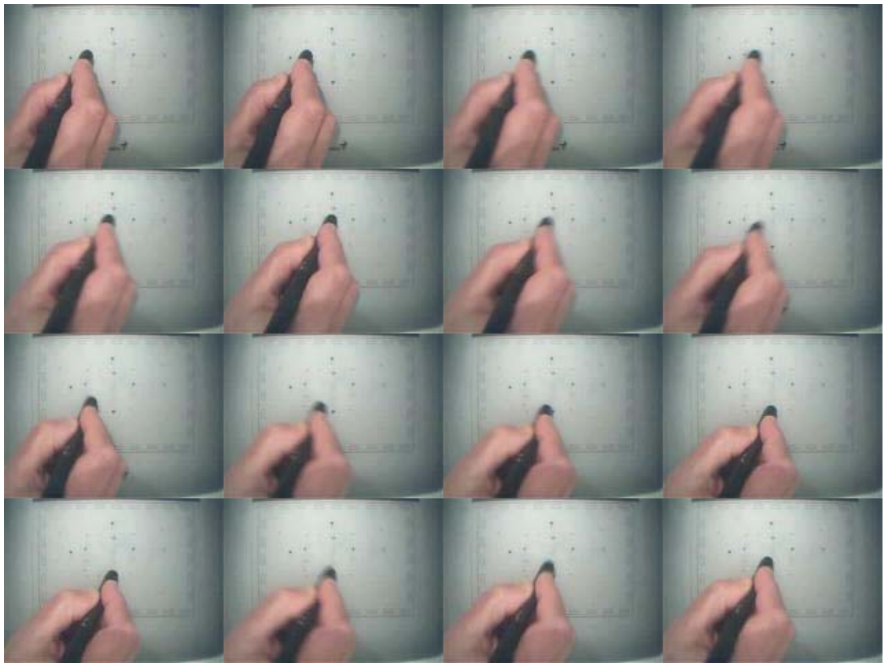

Figure A-4 Example of writing letter (ج) for user 2.