Assaleh, K., Shanableh, T., Fanaswala, M., Amin, F., & Bajaj, H. (2010). Continuous Arabic sign language recognition in user dependent mode. Journal of Intelligent learning systems and applications, 2(01), 19- 27

doi:10.4236/jilsa.2010.21003
Published Online February 2010 (http://www.SciRP.org/journal/jilsa)

Continuous Arabic Sign Language Recognition in User Dependent Mode

K. Assaleh[1], T. Shanableh[2], M. Fanaswala[1], F. Amin[1], and H. Bajaj[1]
[1,2]College of Engineering
[1,2]Department of Electrical Engineering
[2]Department of Computer Science and Engineering
[1,2]American University of Sharjah, UAE
kassaleh@aus.edu

Abstract

Arabic Sign Language recognition is an emerging field of research. Previous attempts at automatic vision-based recognition of Arabic Sign Language mainly focused on finger spelling and recognizing isolated gestures. In this paper we report the first continuous Arabic Sign Language by building on existing research in feature extraction and pattern recognition. The development of the presented work required collecting a continuous Arabic Sign Language database which we designed and recorded in cooperation with a sign language expert. We intend to make the collected database available for the research community. Our system which we based on spatio-temporal feature extraction and hidden Markov models has resulted in an average word recognition rate of 94%, keeping in the mind the use of a high perplexity vocabulary and unrestrictive grammar. We compare our proposed work against existing sign language techniques based on accumulated image difference and motion estimation. The experimental results section shows that the proposed work outperforms existing solutions in terms of recognition accuracy.

Index terms: pattern recognition, motion analysis, image/ video processing and sign language.

1. INTRODUCTION

The growing popularity of vision-based systems has led to a revolution in gesture recognition technology. Vision-based gesture recognition systems are primed for applications such as virtual reality, multimedia gaming and hands-free interaction with computers. Another popular application is sign language recognition, which is the focus of this paper.

There are two main directions in sign language recognition. Glove-based systems use motion sensors to capture gesture data [1],[2] and [3]. While this data is more attractive to work with, the gloves are expensive and cumbersome devices which detract from the naturalness of the human-computer interaction. Vision-based systems, on the other hand, provide a more natural environment within which to capture the gesture data. The flipside of this method is that working with images requires intelligent feature extraction techniques in addition to image processing techniques like segmentation which may add to the computational complexity of the system.

Note that while respectable results have been obtained in the domains of isolated gesture recognition and finger spelling, research on continuous Arabic sign language recognition is non-existent.

The work in [4] developed a recognition system for ArSL alphabets using a collection of Adaptive Neuro-Fuzzy Inference Systems (ANFIS), a form of supervised learning. They used images of bare hands instead of colored gloves to allow the user to interact with the system conveniently. The used feature set comprised lengths of vectors that were selected to span the fingertips' region and training was accomplished by use of a hybrid learning algorithm, resulting in a recognition accuracy of 93.55%. Likewise [5] reported classification results of Arabic sign language alphabets using Polynomial classifiers. The work reported superior results when compared with their previous work using ANFIS on the same dataset and feature extraction techniques. Marked advantages of polynomial classifiers include their computational scalability, less storage requirements and absence of the need for iterative training. This work required the participants to wear gloves with colored tips while performing the gestures to simplify the image segmentation stage. They extracted 30 features involving the relative position and orientation of the fingertips with respect to the wrist and each other. The resulting system achieved 98.4% recognition accuracy on training data and 93.41% on test data.

Sign language recognition of words/gestures as opposed to alphabets depends on analyzing a sequence of still images with temporal dependencies. Hence HMMs are a natural choice for model training and recognition as reported in [6]. Nonetheless, the work in [7] presented an alternative technique for feature extraction of sequential data. Working with isolated ArSL gestures, they eliminate the temporal dependency of data by accumulating successive prediction errors into one image that represents the motion information. This removal of temporal dependency allows for simple classification methods, with less computational and storage requirements. Experimental results using k-Nearest Neighbors and Bayesian classifiers resulted in 97 to 100% isolated gesture recognition. Variations of the work in [7] include the use of block-based motion estimation in the feature extraction process. The resultant motion vectors are used to represent the intensity and directionally of the gestures' motion as reported in [8].

Other sign languages such as American Sign Language have been researched and documented more thoroughly. A common approach in ASLR (American Sign Language Recognition) of continuous gestures is to use Hidden Markov Models as classifier models. Hidden Markov Models are an ideal choice because

they allow modeling of the temporal evolution of the gesture. In part, the success of HMMs in speech recognition has made it an obvious choice for gesture recognition. Research by Starner and Pentland [9] uses HMMs to recognize continuous sentences in American Sign Language, achieving a word accuracy of 99.2%. Users were required to wear colored gloves and an 8-element feature set, comprising hands' positions, angle of axis of least inertia, and eccentricity of bounding ellipse, was extracted. Lastly, linguistic rules and grammar were used to reduce the number of misclassifications.

Another research study by Starner and Pentland [10] dealt with developing a Real-time ASLR system using a camera to detect bare hands and recognize continuous sentence-level sign language. Experimentation involved two systems: first, using a desk mounted camera to acquire video, that attained 92% recognition and second, mounting the camera in the user's cap, which achieved an accuracy of 98%. This work was based on limited vocabulary data, employing a 40-word lexicon. The authors do not present sentence recognition rates for comparison. Only word recognition and accuracy rates are reported.

This paper is organized as follows. Section 2 describes the Arabic sign language database constructed and used in the work. The methodology followed is enumerated in Section 3. The results are discussed in Section 4. Concluding remarks along with a primer on future work in presented in Section 5.

2. THE DATASET

Arabic Sign Language is the language of choice amongst the hearing and speech impaired community in the Middle East and most Arabic speaking countries. This work involves two different databases; one for isolated gesture recognition and another for continuous sentence recognition. Both datasets are collected in collaboration with Sharjah City for Humanitarian Services (SCHS) [11], no restriction on clothing or background was imposed. The first database was compiled for isolated gesture recognition as reported in [7]. The dataset consists of 3 signers acting 23 gestures. Each signer was asked to repeat each gesture a total of 50 times over 3 different sessions resulting in a total of 150 repetitions of each gesture. The gestures are chosen from the greeting section of the Arabic sign language.

The second database is of a relatively high perplexity consisting of an 80-word lexicon from which 40 sentences were created. No restrictions are imposed on grammar or sentence length. The sentences and words pertain to common situations in which handicapped people might find themselves in. The dataset itself consists of 19 repetitions of each of the 40 sentences performed by only one user. The frame rate was set to 25Hz with a spatial resolution of 720x528. The list of sentences is given in Table 1. Note that this database is the first fully labeled and segmented dataset for continuous Arabic Sign Language. The entire database can be made available on request.

No.	Arabic Sentence	English Meaning
1.	ذهبت الى نادي كرة القدم I went to the soccer club	
2.	انا احب سباق السيارات I love car racing	
3.	اشتريت كرة ثمينة I bought an expensive ball	
4.	يوم السبت عندي مباراة كرة قدم On Saturday I have a soccer match	
5.	في النادي ملعب كرة قدم	There is a soccer field in the club

6.	غدا سيكون هناك سباق دراجات	There will be a bike racing tomorrow
7.	وجدت كرة جديدة في الملعب	I found a new ball in the field
8.	كم عمر اخيك؟	How old is your brother?
9.	اليوم ولدت امي بنتا	My mom had a baby girl today
10.	اخي لا يزال رضيعا	My brother is still breast feeding
11.	ان جدي في بيتنا	My grandfather is at our home
12.	اشترى ابني كرة رخيصة	Mu kid bought an inexpensive ball
13.	قرأت اختي كتابا	My sister read a book
14.	ذهبت امي الى السوق في الصباح	My mother went to the market this morning
15.	هل اخوك في البيت؟	Is your brother home?
16.	بيت عمي كبير	My brother's house is big
17.	سيتزوج اخي بعد شهر	In one month my brother will get married
18.	سيطلق اخي بعد شهرين	In two months my brother will get divorced
19.	اين يعمل صديقك؟	Where does your friend work?
20.	اخي يلعب كرة سلة	My brother plays basketball
21.	عندي أخوين	I have two brothers
22.	ما اسم ابيك؟	What is your father's name?
23.	كان جدي مريضا في الامس	Yesterday my grandfather was sick
24.	مات ابي في الامس	Yesterday my father died
25.	رأيت بنتا جميلة	I saw a beautiful girl
26.	صديقي طويل	My friend is tall
27.	انا لا آكل قبل النوم	I do not eat close to bedtime
28.	اكلت طعاما لذيذا في المطعم	I ate delicious food at the restaurant
29.	انا احب شرب الماء	I like drinking water
30.	انا احب شرب الحليب في المساء	I like drinking milk in the evening
31.	انا احب اكل اللحم اكثر من الدجاج	I like eating meat more than chicken
32.	اكلت جبنة مع عصير	I ate cheese and drank juice
33.	يوم الاحد القادم سيرتفع سعر الحليب	Next Sunday the price of milk will go up
34.	أكلت زيتونا صباح الامس	Yesterday morning I ate olives
35.	ساشتري سيارة جديدة بعد شهر	I will buy a new car in a month
36.	هو توضأ ليصلي الصبح	He washed for morning prayer
37.	ذهبت الى صلاة الجمعة عند الساعة العاشرة	I went to Friday prayer at 10:00 o'clock
38.	شاهدت بيتا كبيرا بالتلفاز	I saw a big house on TV
39.	في الامس نمت عند الساعة العاشرة	Yesterday I went to sleep at 10:00 o'clock
40.	ذهبت الى العمل في الصباح بسيارتي	I went to work this morning in my car

Table 1. List of Arabic sentences with English translation used in the recognition system.

A required step in all supervised learning problems is the labeling stage where the classes are explicitly marked for the classifier training stage. For continuous sentence recognition, not only do the sentences have to be labeled but the individual boundaries of the gestures that make up that sentence have to be explicitly demarcated. This is a time-consuming and repetitive task. Conventionally, a portion of the data is labeled and used as 'bootstrap' data for the classifier which can then learn the remaining boundaries. For the purposes of creating a usable database, a segmented and fully labeled dataset was created in the Georgia Tech Gesture Recognition toolbox (GT^2K) format [12]. The output of this stage is a single master label file (MLF) that can be used with the GT^2K and HTK Toolkits.

3. FEATURE EXTRACTION

In this section we introduce a feature extraction technique suitable for continuous signing. We also examine some of the existing techniques and adapt them to our application for comparison reasons.

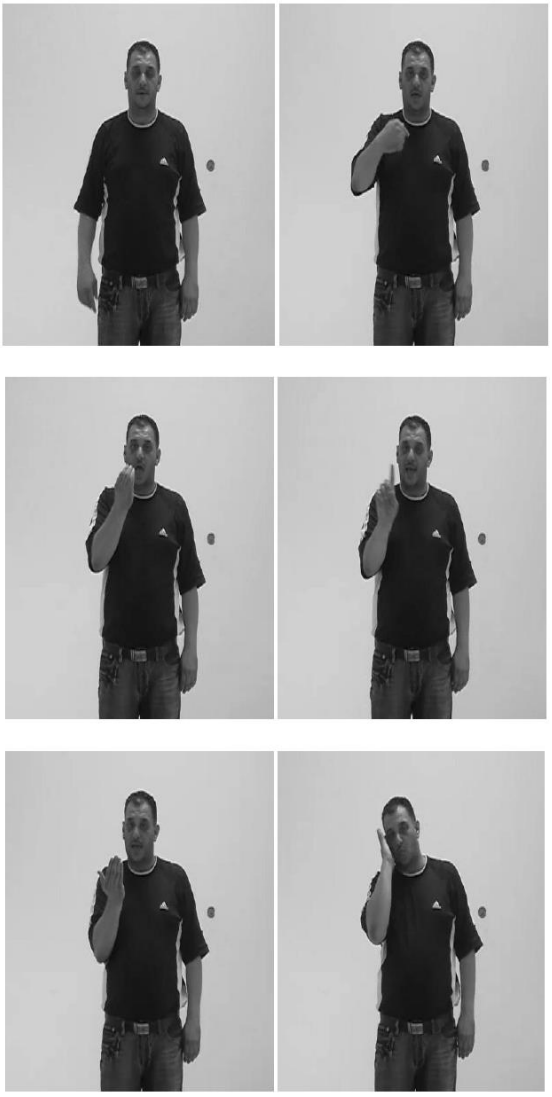
3.1 Proposed feature extraction

The most crucial stage of any recognition task is the selection of good features. Features that are representative of the individual gestures are desired. Shanableh *et. al.* [7] demonstrated in their earlier work on isolated gesture recognition that the two-tier spatial-temporal feature extraction scheme results in a high word recognition rate close to 98%. Similar extraction techniques are used in our continuous recognition solution.

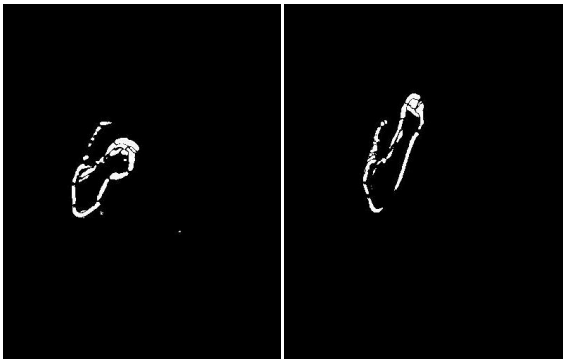
First, to represent the motion that takes place as the expert signs a given sentence, pixel-based differences of successive images are computed.

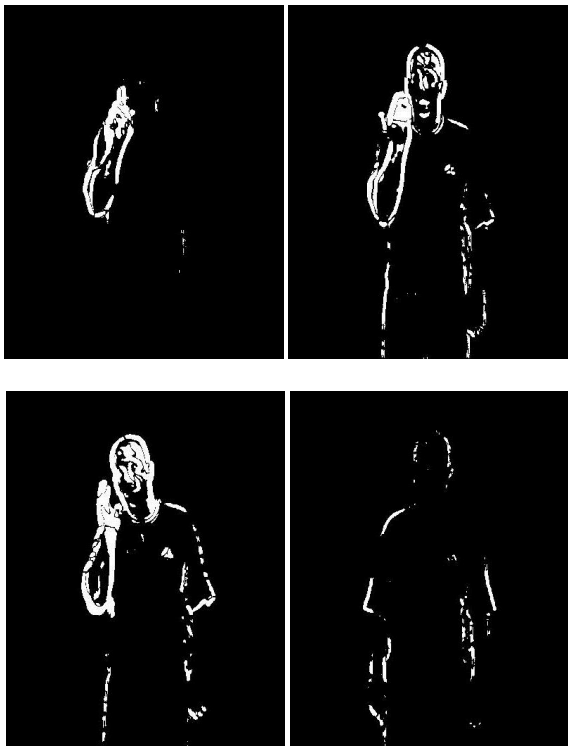
It can be justified that the difference between two images of similar background results in an image that only preserves the motion between the two images. These image differences are then converted into binary images by applying an appropriate threshold. A threshold value of $\mu+x\sigma$ is used where μ is the mean pixel intensity of the image difference, σ is the corresponding standard deviation and x is a weighting parameter which was empirically determined based on subjective evaluation whose criteria was to retain enough motion information and discarding the noisy data.

Figure 1 shows an example sentence with thresholded image differences. Notice that the example sentence is temporally downsampled for illustration purposes.



(a) An image sequence denoting the sentence ‘I do not eat close to bed time’.





(b) Thresholded image differences of the image sequence in part a

Figure 1. An example sentence and its motion representation.

Next, a frequency domain transformation such as the Discrete Cosine Transform (DCT) is performed on the binary image differences.
The 2-D Discrete Cosine Transformation (DCT) given by [13]:

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

Where NxM are the dimensions of the input image 'f' and 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.

In Figure 4, it is apparent that the DCT transformation of a thresholded image difference results in energy compaction where most of the image information is represented in the top left corner of the transformed image.
Subsequently, zig-zag scanning is used to select only a required number of frequency coefficients. This process is also known as zonal coding. The number of coefficients to retain or the DCT cutoff is elaborated upon in the experimental results section.

Figure 4 – Discrete Cosine Transform Coefficients of a thresholded image difference.

These coefficients obtained in a zig-zag manner make up the feature vector that is used in training the classifier.

3.2 Adapted feature extraction solutions

For completeness, we compare our feature extraction solution to existing work on Arabic sign language recognition. Noteworthy are the Accumulated Differences (ADs) and Motion Estimation (ME) approaches to feature extraction as reported in [8,17]. In this section we provide a brief review of each of mentioned solutions and explain who it can be adapted to our problem of continuous Arabic sign language recognition

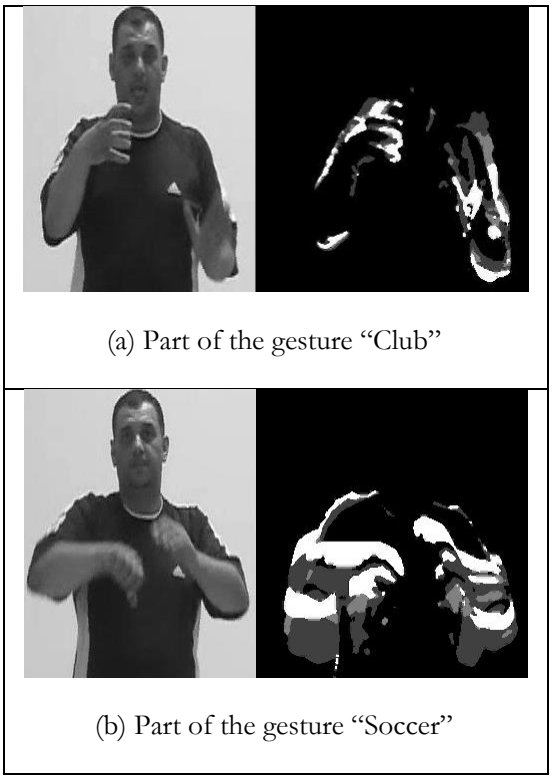
3.2.1 Accumulated differences solution

The motion information of an isolated sign gesture can be computed from the temporal domain of its image sequence through successive image differencing. Let $I_{g,i}^{(j)}$ denote image index j of the i^{th} repetition of a gesture at index g. The Accumulated Differences (ADs) image can be computed by:

$$AD_{g,j} = \sum_{j=1}^{n-1} \partial_j \left(\left| I_{g,j}^{(j)} - I_{g,i}^{(j-1)} \right| \right) \qquad (2)$$

Where n is the total number of images in the i^{th} repetition of a sign at index g. ∂_j is a binary threshold function of the j^{th} frame.

Note that the ADs solution cannot be directly applied to continuous sentences (as opposed to isolated sign gestures). This is so because the gesture boundaries in a sentence are unknown, thus one solution is to use an overlapping sliding window approach in which a given number of video frame differences are accumulated into one image regardless of gesture boundaries. The window is shift by one video frame at a time. In the experimental results section we experiment with various window sizes. Examples of such accumulated differences are shown in Figure 5 with a window size of 8 video frames. Notice that the ADs capture the frame difference between the current and previous video frames and it also accumulates the frame differences from the current window as well.



(a) Part of the gesture "Club"

(b) Part of the gesture "Soccer"

Figure 5. Example Accumulated Differences images using an overlapping sliding window of size 8.

Once the ADs image is computed it is then transformed into the DCT domain as described previously. The DCT coefficients are Zonal coded to generate the feature vector.

3.2.2. Motion Estimation solution

The motion of a video-based sign gesture can also be tracked by means of Motion Estimation (ME). One well known example is the block-based ME in which the input video frames are divided into non-overlapping blocks of pixels. For each block of pixels, the motion estimation process will search through the previous video frame for the "best match" area within a given search range. The displacement, in terms of pixels, between the current block and its best match area in the previous video frame is represented by a motion vector

Formally, let *C* denote a block in the current video frame with *bxb* pixels at coordinates *(m,n).* Assuming that the maximum motion displacement is *w* pixel per video frame then the the motion estimation process will find the best match area *P* within the *(b+2w)(b+2w)* distinct overlapping *bxb* blocks of the previous video frame. An area in the previous video frame that minimizes a certain distortion measure is selected as the best match area. A common distortion measure is the mean absolute difference given by:

$$M(\Delta x, \Delta y) = \frac{1}{b^2} \sum_{m=1}^{b} \sum_{n=1}^{b} \left| C_{m,n} - P_{m+\Delta x, n+\Delta y} \right|, \quad -w \le \Delta x, \Delta y \le w \qquad (3)$$

Where $\Delta x, \Delta y$ refer to the spatial displacement in between the pixel coordinates of C and the matching area in the previous image. Other distortion measures can be used such as mean squared error, cross correlation functions and so forth. Further details on motion estimation can be found in [18] and references within.

The motion vectors can then be used to represent the motion the occurred between two video frames. These vectors are used instead of the thresholded frame differences. In [8] it was proposed to rearrange the *x* and y components of the motion vectors into two intensity images. The two images are then concatenated to generate one representation of the motion that occurred between two video frames. Again, once the concatenated image is computed it is then transformed into the DCT domain as described previously. The DCT coefficients are Zonal coded to generate the feature vector.

4. CLASSIFICATION

For conventional data, naïve Bayes classification provides the upper bound for the best classification rates. Since sign language varies in both spatial and temporal domains, the extracted feature vectors are sequential in nature and hence simple classifiers might not suffice. There are two main approaches to dealing with sequential data. The first method aims to combine the sequential feature vectors using a suitable operation into a single feature vector. A detailed account of such procedures is outlined in [14]. One such method involves concatenating sequential feature vectors using a sliding window of optimal length to create a single feature vector. Subsequently, classical supervised learning techniques such as maximum-likelihood estimation (MLE), linear discriminants or neural networks can be used. The second approach makes explicit use of classifiers that can deal with sequential data without concatenation or accumulation, such an approach is used in this paper.

While the field of gesture recognition is relatively young, the related field of speech recognition is well established and documented. Hidden Markov Models are the classifier of choice for continuous speech recognition and lend themselves suitably for continuous sign language recognition too. As mentioned in [15], a HMM is a finite-state automaton characterized by stochastic transitions in which the sequence of states is a Markov chain. Each output of an HMM corresponds to a probability density function. Such a generative model can be used to represent sign language units (words, sub-words etc).

5.1 Hidden Markov Models (HMMs)

HMM is a statistical model used to study time varying sequences. The system being modeled is assumed to be a Markov process with 'hidden' or unknown parameters to be determined from observable sequences. An HMM model can be characterized by the following elements:

- N, the total number of states in the model.

The states or sets of states in a Hidden Markov Model have some physical significance to the process being studied. These states are interconnected in a manner specified by the model. In the following discussion we will denote them as {1,2,…..N} and denote the state at time t as q_t.

- M, the number of observations per state. The observations correspond to the physical output of the system being modeled.
- A, the state transition probabilities $A = \{a_{ij}\}$, where

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \le i, j \le N \qquad (4)$$

$a_{ij} > 0$ for all i, j for the special case where any state can reach any other state in a single step. For all other cases, $a_{ij} = 0$ for one or more (i, j) pairs.

- B, the output sequence probabilities $B = \{b_j(k)\}$, where

$$b_j(k) = P(v_k at\ t | q_t = S_j) \quad 1 \le j \le N\ and\ 1 \le k \le M \qquad (5)$$

- $\boldsymbol{\pi}$, the initial state distribution

$\pi = \{\pi_i\}$, where

$$\pi_i = P(q_t = S_i) \quad 1 \le i \le N. \qquad (6)$$

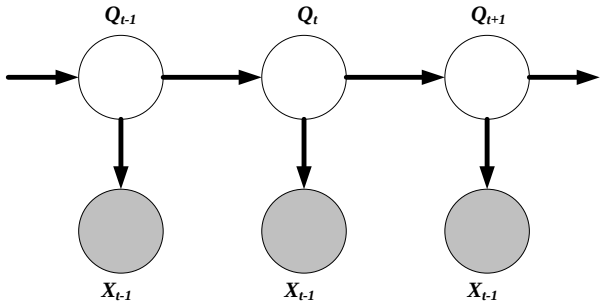


Figure 6- Visualization of a Hidden Markov Model

Given the form of the HMM described above, we are faced with three basic issues for the model before the theory is applied to real-world problems. The problems are described below in a brief fashion following which there is some additional detail if the reader wishes to know more.

5.1.1 The Evaluation Problem

We seek a solution to the problem of efficiently computing the probability of the observation sequence given the model. Mathematically, we seek to compute $P(\mathbf{O} | \lambda)$, given an observation sequence $\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 ... \mathbf{o}_T)$ and a model $\lambda = (A, B, \pi)$. This problem can be thought of as evaluating how well an observation sequence matches a particular problem. This stage is primarily used in recognition of the model which generated the observation sequence by a scoring system.

A naïve method of solving this problem would be to compute the joint probability of the observation sequence and a particular state sequence given a model and then summing over all possible state sequences. We seek the probability

$$\sum_{allq} P(\mathbf{O}, \mathbf{q} | \lambda) = \sum_{allq} P(\mathbf{O} | \mathbf{q}, \lambda) P(\mathbf{q} | \lambda) \qquad (7)$$

$$\sum_{allq} P(\mathbf{O},\mathbf{q}\mid\lambda) = \sum_{q_1,q_2....q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1q_2} b_{q_2}(\mathbf{o}_2).....a_{q_{T-1}q_T} b_{q_T}(\mathbf{o}_T) \qquad (8)$$

But this is computationally prohibitive and hence an inductive method called the Forward Procedure is used.

The Forward procedure is formulated below where the forward variable is defined as $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2....\mathbf{o}_t, q_t = i\mid\lambda)$, which is the joint probability of the partial observation sequence $\mathbf{o}_1\mathbf{o}_2....\mathbf{o}_t$ (until time t) and being in state i at time t.

Step 1: Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1) \qquad for\ 1\le i\le N \qquad (9)$$

Step 2: Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}) \qquad for\ 1\le i\le T-1,\ 1\le j\le N \qquad (10)$$

Step 3: Termination

$$P(\mathbf{O}\mid\lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad (11)$$

5.1.2 The Decoding Problem

We seek to choose the hidden state sequence $\mathbf{q} = (q_1q_2....q_T)$ that optimally explains the observation sequence $\mathbf{O} = (\mathbf{o}_1\mathbf{o}_2....\mathbf{o}_T)$ given the model $\lambda = (A,B,\pi)$. Decoding can be used to learn about the model structure, finding optimal state sequences or to get average statistics of individual states. It can also be used to segment each of the gesture training sequences into states.

The decoding problem is solved by formulating it as an optimization problem. There are various choices for optimality criteria but the most widely used criterion is to maximize $P(\mathbf{q}\mid\mathbf{O},\lambda)$. The Viterbi algorithm is the formal technique used to find the best state sequence by maximizing the above criterion.

$\delta_t(i) = \max_{q_1,q_2....q_{t-1}} P[q_1q_2....q_{t-1}, q_t = i, \mathbf{o}_1\mathbf{o}_2.....\mathbf{o}_t\mid\lambda]$, which the highest probability along a particular state sequence accounting for the first 't' observations and ending in state i.

By induction, we can also define,

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}) \qquad (12)$$

The arguments that maximize the above at each $t \text{ and } j$ are recorded in the array $\psi_t(j)$

Step 1: Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1) \qquad for\ 1 \le i \le N \tag{13}$$

$$\psi_1(i) = 0$$

Step 2: Recursion

$$\delta_t(j) = \left[\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij}\right] b_j(\mathbf{o}_t) \quad for\ 2 \le t \le T,\ 1 \le j \le N \tag{14}$$

$$\psi_t(j) = \left[\arg\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij}\right] \quad for\ 2 \le t \le T,\ 1 \le j \le N \tag{15}$$

Step 3: Termination

$$P^* = \max_{1 \le i \le N} \delta_T(i) \tag{16}$$

$$q^* = \arg\max_{1 \le i \le N} \delta_T(i) \tag{17}$$

Step 4: Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \ldots.1 \tag{18}$$

5.1.3 The Training Problem:

We seek to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize the probability of the observation sequence. Mathematically, we seek to maximize $P(\mathbf{O} \mid \lambda)$ by adjusting the model parameters. This is done by choosing the model parameters such that we maximize the likelihood, $P(\mathbf{O} \mid \lambda)$ locally using iterative procedures or a gradient search. The conventional method of Baum-Welch re-estimation utilizes an iterative procedure of expectation-maximization. For continuous observation densities, the re-estimation formulas are:

$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}, \tag{19}$$

where $\bar{c}_{jk}$ are the mixture coefficients for the kth mixture in state j, and M is the number of coefficients.

$$\bar{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\ \mathbf{o}_T}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{20}$$

where $\bar{\boldsymbol{\mu}}_{jk}$ is an estimate of the mean of the k[th] mixture in state j.

$$\bar{\mathbf{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\ (\mathbf{o}_T - \boldsymbol{\mu}_{jk})(\mathbf{o}_T - \boldsymbol{\mu}_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)}, \tag{21}$$

where $\bar{\mathbf{U}}_{jk}$ is an estimate of the co-variance matrix of the k[th] mixture in state j.

For a more rigorous treatment of Hidden Markov Models and their applications, the reader is encouraged to refer to the work presented in [16].

5.2 HMM implementation

The implementation of an HMM framework was carried out using the Georgia Tech Gesture Recognition Toolkit (GT^2K) which serves as a wrapper for the more general Hidden Markov Model Toolkit (HTK). The GT^2K version used was a UNIX based package. HTK is the de-facto standard in speech recognition application using HMM's.

A logical step in proceeding from isolated gesture recognition would be connected gesture recognition. This can be simulated by concatenating individual gestures into artificial sentences. Intuitively, one would expect better results for connected gesture recognition as opposed to continuous gesturing. This is because concatenated gestures do not suffer from the altered spatial gesturing that occurs as gestures are signed continuously without pauses.

The first database consisting of isolated gestures was used to create concatenated sentences of varying length. These sentences were created without any consideration of whether the constructed sentence held any meaning or grammatical structure. This concatenated data was divided into a training set and a testing set comprised of 70% and 30% of the total data respectively. The GT^2K Toolkit was then used to perform recognition based on individual words as the basic unit of Arabic sign language. While concatenation is not the aim of this work, the results obtained provide a valuable benchmark for subsequent experiments with continuous sentence signing. An average of 96% sentence recognition and 98% word recognition was obtained on the concatenated testing dataset. The word recognition rate is comparable to previous work in ArSL [7] using similar feature extraction schemes. It would be prudent to note that due to the nature of concatenation, the boundary between gestures is prominent and this might account for the high sentence recognition rate.

The second database was then used to perform continuous sentence recognition. This was also performed with the help of GT^{2}K. This data is also divided into a training (70%) and testing set (30%). An average of 75% sentence recognition and 94% word recognition was obtained on the testing set. A detailed analysis of the various associated parameters is given in the experimental results section.

5. Experimental RESULTS

There are several parameters that affect the recognition rates in continuous sign language recognition. Namely, the sections below discuss the effect of varying the number of hidden states, number of guassian mixtures, length of feature vectors and the threshold used for binarizing the image differences. Unless otherwise stated, the length of the feature vectors used throughout the experiments is 100 DCT coefficients.

The following results are based on the word and sentence recognition rates. The former is computed through the following equation:

$$Accuracy_{word} = 1 - \frac{D + S + I}{N} \qquad (22)$$

Where D is the number of deletions, S is the number of substitutions, I is the number of insertions, and N is the total number of words. On the other hand, sentence recognition rate is the ratio of the correctly recognized sentences to the total number of sentences. Correctness in this case entails correct recognition of all the words constituting the sentence without any insertions, substitutions, or deletions.

5.1. Number of Hidden States

In Figure 7, the effect of increasing the number of hidden states in the HMM topology on sentence and word recognition rates is examined.

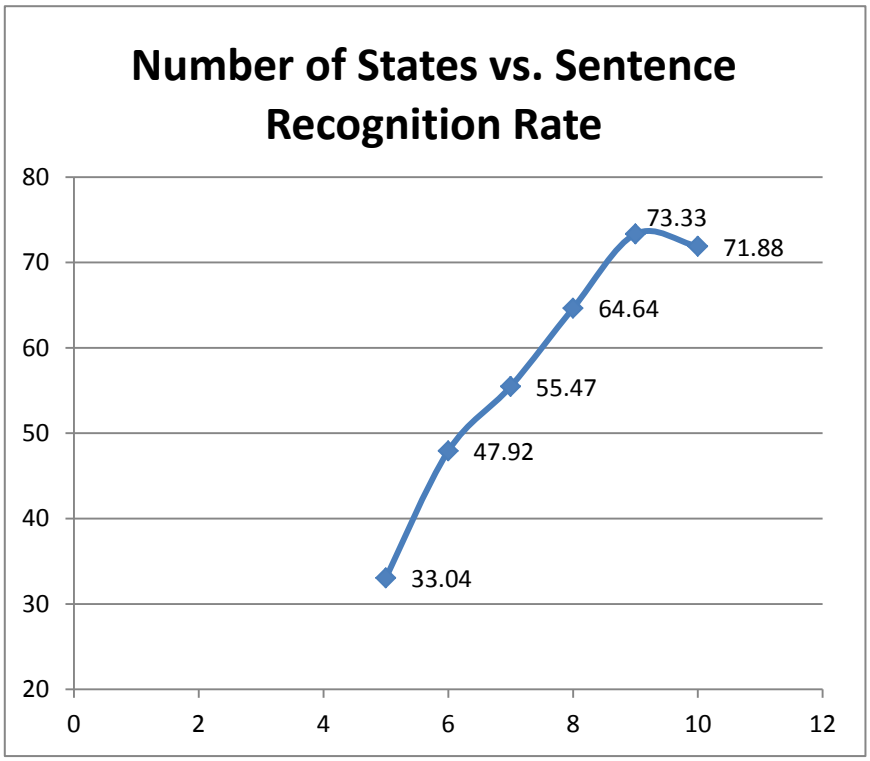


Figure 7 - The effect of number of states on the sentence recognition rate (3 Gaussian mixtures are used)

An increasing trend with the recognition rates is observed as the number of states is increased to a certain number and then the classification rates saturates with a subsequent drop. As the number of states in the Hidden Markov Model is increased, we are in effect increasing the degrees of freedom allowed in modeling the data. Working with video data sampled at 25 frames per second, the classification rate increased to a maximum at nine states. The saturation in recognition accuracy is attributed to the fact that certain gestures do not extend for a long time duration and are only represented by few frame differences. The increase in number of states only serves to increase computation time while adding redundant data that does not contribute to the classification rate.

5.2. Length of the feature vector

Figure 8 shows the recognition rates for increasing the number of DCT coefficients within the feature vector.

It is expected that the increase in feature vector size be accompanied by a corresponding increase in recognition rates. This is due to the fact that each DCT coefficient is uncorrelated with other coefficients and hence no redundant information is present in increasing coefficients. Experimental results shown in Figure 8 show a general increase in recognition rates as the number of DCT coefficients is increased.

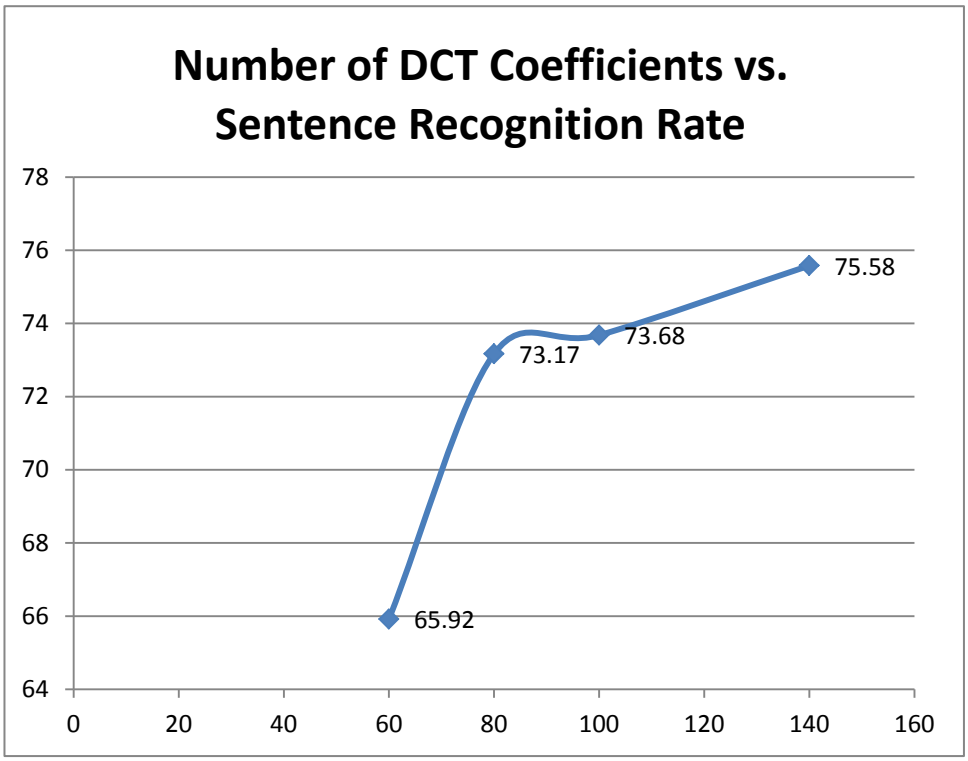


Figure 8 – The effect of length of the feature vector on sentence recognition rates (3 Gaussian mixtures are used with a HMM topology using 9 states).

The trend shows that any increase in recognition accuracy beyond 100 coefficients is only slightly significant. The increase in computation time is however a limiting factor in increasing the length of the feature vector indiscriminately.

5.3. Number of Gaussian Mixtures

The effect of increasing the number of Gaussian mixtures is shown in Figure 9 and 10.
Gaussian mixtures are used to model the emission probability densities of each state of a continuous Hidden Markov Model.

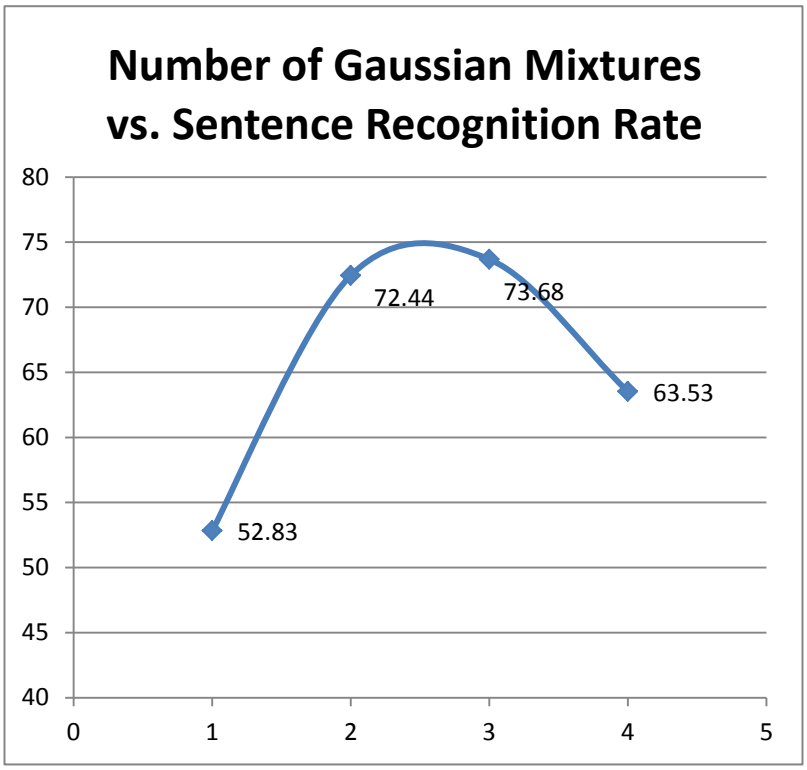


Figure 9 - The effect of the number of gaussian mixtures on the sentence recognition rate (HMM topology using 9 states).

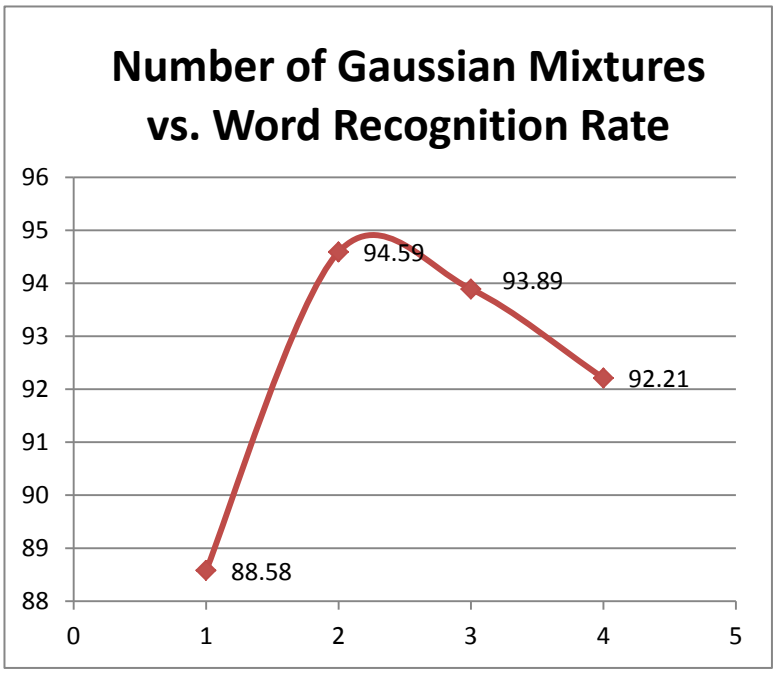


Figure 10 - The effect of the number of gaussian mixtures on the word recognition rate (HMM topology using 9 states).

For multi-dimensional data like the long feature vectors used in this work it is desirable to have a number of Gaussian mixtures so that any emission pdf can be effectively fit. Increasing the number of Gaussian mixture shows substantial improvement in recognition rates. The results depict a general increase in recognition rates as the mixtures are increased. However, the authors feel that the limitation in collecting large amounts of data does not allow the use of more mixtures.

5.4. Choice of Threshold

In the feature extraction process, the image differences are thresholded into binary images based on a threshold of $\mu+x\sigma$, where μ is the mean pixel intensity of the image difference, σ is the corresponding standard deviation and x is a weight parameter. Results are shown in Figure 11 and 10 for different values of the weight parameter.

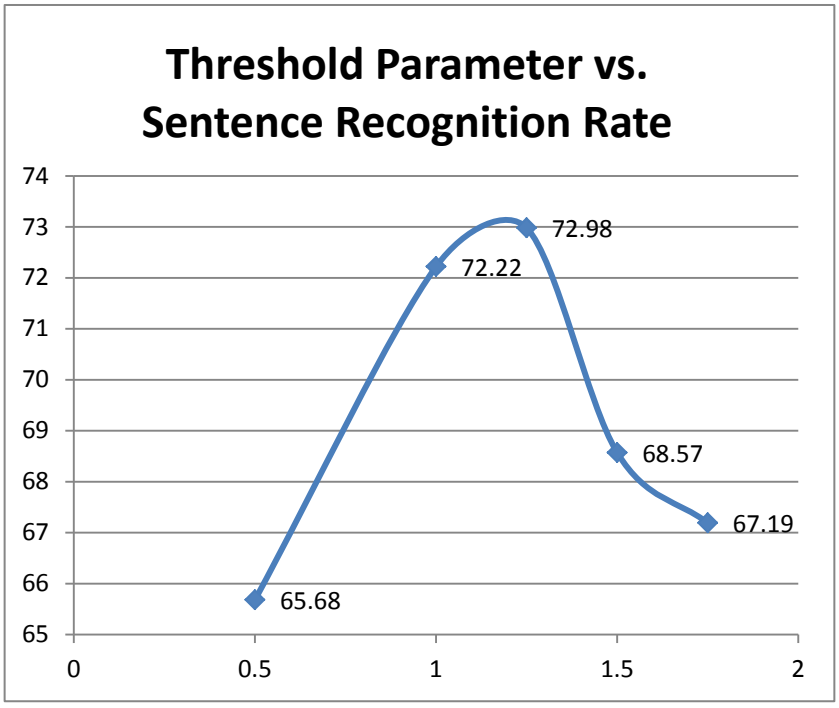


Figure 11 - The effect of the weighting factor on the sentence recognition rate (3 Gaussian mixtures are used with a HMM topology using 9 states).

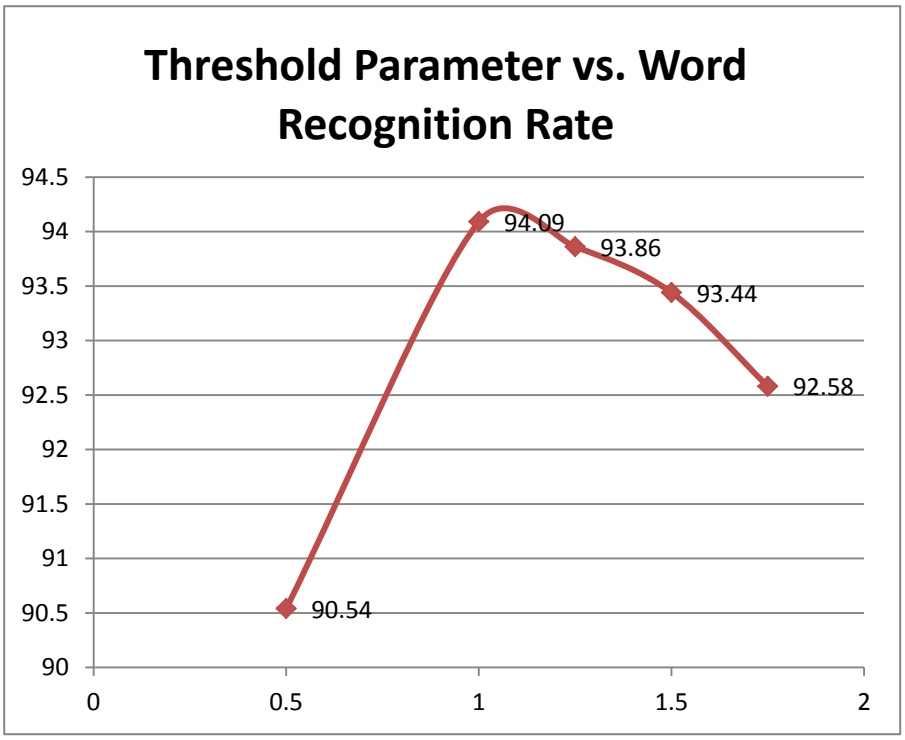


Figure 12 - Effect of the weighting factor on the word recognition rate (3 Gaussian mixtures are used with a HMM topology using 9 states).

The recognition accuracy peaks at a weighting parameter value between 1 and 1.25. A subjective comparison of the thresholded image differences shows that these parameter values retain most of the motion information whilst discarding spurious information such as small stray shifts in clothing, illumination and the like.

Lastly as mentioned in Section 3.2 above, we compare our solution against existing work on Arabic sign language recognition. Namely we consider both the ADs and ME approaches to feature extraction. Table 2 summaries the sentence and word recognition rates using various feature extraction solutions. The experimental parameters are similar to those used in Figure 12.

Feature extraction approach	Sentence recognition rate	Word recognition rate
ADs with an overlapping Sliding window of size 4.	64.1%	91.0%
ADs with an overlapping Sliding window of size 7.	65.2%	90.6%
ADs with an overlapping Sliding window of size 10.	68%	93.71%
Motion estimation	67.9%	92.9%
Proposed solution	73.3%	94.39%

Table 2. Comparisons with existing feature extraction solutions.

The recognition results presented in the table indicate that the proposed solution provides the highest sentence and word recognition rates. The ADs with the overlapping sliding window approach was not advantageous. Intuitively the ADs image puts the difference between 2 video frames into context by accumulating future frame differences to it. However in HMMs temporal information is preserved and therefore extracting feature vectors from video frame differences without accumulating them will suffice. It is also worth mentioning that increasing the window size beyond 10 frames did not further enhance the recognition rate.

In the ME approach, the image block size and the search range are set to 8x8 pixels which is a typical setting in video processing. The resultant recognition rates are comparable to the ADs approach. Note that ME techniques do not entirely capture the true motion of a video sequence. For instance with block-based search techniques object rotations are not captured as good as translational motion. Therefore the recognition results are inferior to the proposed solution.

6. Conclusion and Future Work

The work outlined in this paper is an important step in this domain as it represents the first attempt to recognize continuous Arabic sign language. The work entailed compiling the first fully labeled and segmented dataset for continuous Arabic Sign Language which we intend to make public for the research community. The average sentence recognition rate of 75% and word recognition rate of 94% are obtained using a natural vision-based system with no restrictions on signing such as the use of gloves. Furthermore, no grammar is imposed on the sentence structure which makes the recognition task more challenging. The use of grammatical structure can significantly improve the recognition rate by alleviating some types of substitution and insertion errors. In the course of training, the dataset was plagued by an unusually large occurrence of insertion errors. This problem was mitigated by applying a detrimental weight for every insertion error which was incorporated into the training stage. As a final comment, the perplexity of the dataset is large compared to other work in related fields.

Future work in this area aiming to secure higher recognition rates might require a sub-gesture (analogous to phonemes in speech recognition) based recognition system. Such a system would also serve to alleviate the motion-epenthesis effect which is similar to the co-articulation effect in speech recognition. Such a system would also require a psycho-linguistic study on the structure of Arabic sign language.

The work presented in this paper is also limited to the user-dependent domain. The feature extraction techniques used in this work are scalable towards user-independent applications.

Finally, the frequency domain transform coefficients used as features perform well in concatenated gesture recognition. The average word recognition rate is also sufficiently high with an average of 94%. The authors feel that geometric features might be used in addition to the existing feature to create an optimum feature set. These geometric features would require the use of segmentation techniques but might result in a substantial increase in sentence recognition rates.

Acknowledgement

The authors would like to acknowledge the invaluable help of Mr. Salah Odeh from the Sharjah City for Humanitarian Services in the construction of the databases.

References

[1] J. S. Kim,W. Jang, and Z. Bien, “A dynamic gesture recognition system for the Korean sign language (KSL),” IEEE Trans. Syst., Man, Cybern. B, vol. 26, pp. 354–359, Apr. 1996.

[2] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima, “The recognition algorithm with noncontact for Japanese sign language using morphological analysis,” Proc. Gesture Workshop, pp. 273–284, 1997.

[3] S. S. Fels and G. E. Hinton, “Glove-talk: A neural network interface between a data-glove and a speech synthesizer,” IEEE Trans. Neural Networks, vol. 4, pp. 2–8, Jan. 1993.

[4] O. Al-Jarrah, A. Halawani, “Recognition of gestures in Arabic sign language using neuro-fuzzy systems,” Artificial Intelligence, 133(1-2), pp. 117-138, December, 2001.

[5] K. Assaleh and M. Al-Rousan, “Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers,” EURASIP Journal on Applied Signal Processing, 2005(13):2136-2146, 2005.

[6] M. AL-Rousan , K. Assaleh and A. Tala'a , “Video-based signer-independent Arabic sign language recognition using hidden Markov models,” Applied Soft Computing, 9(3), June, 2009.

[7] T. Shanableh, K. Assaleh and M. Al-Rousan, “Spatio-Temporal Feature-Extraction Techniques for Isolated Gesture Recognition in Arabic Sign Language,” IEEE Trans. on Systems, Man and Cybernetics Part B, 37(3):641-650, 2007.

[8] T. Shanableh and K. Assaleh, “Telescopic Vector Composition and polar accumulated motion residuals for feature extraction in Arabic Sign Language recognition,” EURASIP Journal on Image and Video Processing, vol. 2007, Article ID 87929, 10 pages, 2007. doi:10.1155/2007/87929.

[9] T. Starner, J. Weaver, A. Pentland. “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(12): 1371-1375, 1998.

[10] T. Starner, J. Weaver and A. Pentland. “A Wearable Computer-Based American Sign Language Recogniser,” in Personal and Ubiquitous Computing, 1997.

[11] Sharjah City for Humanitarian Services (SCHS), website: http://www.sharjah-welcome.com/schs/about/

[12] T. Westeyn, H. Brashear and T. Starner, “Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition,” Proc. International Conference on Perceptive and Multimodal User Interfaces. Vancouver, B.C., November 2003

[13] K. R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications," Academic press, ISBN 012580203X, Aug 1990.

[14] T. Dietterich, "Machine Learning for Sequential Data: A Review," In T. Caelli (Ed.) Structural, Syntactic, and Statistical Pattern Recognition; Lecture Notes in Computer Science, Vol. 2396, pp. 15-30, Springer-Verlag, 2002.

[15] B. Gold and N. Morgan, "Speech and Audio Signal Processing," John Wiley & Sons Inc, ISBN 0471351547, July, 1999.

[16] L. Rabiner, B. Juang, "Fundamentals of Speech Recognition," New Jersey: Prentice-Hall Inc., 1993.

[17] F.-S. Chen, C.-M. Fu and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," Image and Vision Computing, 21(8), pp. 745–758, 2003.

[18] M. Ghanbari, "Video coding: an introduction to standard codecs," Second edition, The Institution of Engineering and Technology, 2003.