2006

# Implementation of image processing approach to translation of ASL finger-spelling to digital text

Divya Mandloi

## Library Rights Statement

In presenting the thesis *Implementation of Image Processing Approach to Translation of ASL Finger-spelling to Digital Text* in partial fulfillment of the requirements for an advanced degree at the Rochester Institute of Technology, I agree that the Library shall make it freely available for inspection. I further agree that permission for copying as provided for by the Copyright Law of the U.S. (Title 17, U.S. Code) of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

I hereby grant permission to the RIT Library to copy my thesis for scholarly purposes.

_____

Divya Mandloi

_____

Date

# IMPLEMENTATION OF IMAGE PROCESSING APPROACH TO TRANSLATION OF ASL FINGER-SPELLING TO DIGITAL TEXT

BY

DIVYA MANDLOI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

TELECOMMUNICATIONS ENGINEERING TECHNOLOGY

ROCHESTER INSTITUTE OF TECHNOLOGY

January 2006

MASTER OF SCIENCE THESIS

OF

DIVYA MANDLOI

APPROVED:

Thesis Committee

Major Professor

_____

_____

_____

ROCHESTER INSTITUTE OF TECHNOLOGY

January 2006

**Abstract**

This thesis describes the ongoing development of an image processing technique for the translation of American Sign Language (ASL) finger-spelling to text. The present analysis is the phase one of a broader project, the Sign2 Project, which is focused on a complete technological approach to the translation of ASL to digital audio and/or text. The methodology adopted in this analysis employs a gray-scale image processing technique to convert the American Sign Language finger-spelling to text. It attempts to process static images of the subject considered, and then matches them to a statistical database of pre-processed images to ultimately recognize the specific set of signed letters. This phase of the Sign2 Project considers the hand of the subject alone and not the entire subject, as its scope is restricted to recognizing the finger spelling and not the American Sign Language as a whole. Since the approach taken in this analysis is vision-based, the amount of processing is minimized as compared to other approaches and hence projects itself as a viable technique to be implemented in real time systems. Devices like kiosks and PDAs can incorporate this technology to enable communication between the hearing and non-hearing individuals who are geographically placed apart with the least possible run times which is mandatory for real-time systems. In this investigation, I intend to describe the approach to the phase one problem and demonstrate the results thus derived, where several words are distinguished and recognized with a fairly high degree of reliability.

**Keywords**

image processing, sign language, ASL, finger-spelling, linguistics, communication

To
Ma, Papa and Tapan

**Acknowledgments**

First of all I would like to thank God for bringing me this far in my personal and academic pursuits and for being my constant guide and strength.

Secondly, I would like to thank my advisor, Dr. Chance M. Glenn for letting me be a part of his vision. I am most grateful to him for his intellectual inputs and personal encouragement during the research. His diligent support has helped me complete the writing of this dissertation and the challenging research that was behind it.

I would also like to thank my graduate advisor, Dr. Warren Koontz for lending a patient ear through all the ups and downs of this research work. I am also grateful to him for reading through my dissertation many times and helping me put it together.

The Laboratory for Advanced communications Technology (LACT) and Graphic Design Lab at College of Applied Science and Technology provided the constructive environment necessary for undertaking research. I am also indebted to all my friends who volunteered by patiently signing letters and for contributing their time and efforts to my work.

Last, but not the least, I would also like to thank my research partners and friends, Kanthi Sarella and Muhammed Jamal for sharing sleepless nights with me working on the project and for adding an element of madness to the intense hours of work.

# Table of Contents

**List of Tables**

**List of Figures**

# Chapter 1
# Introduction

According to statistics gathered by National Center for Health Statistics in 1994 [1], 38 to 140 persons out of every 1000 persons in the United States have hearing loss problems. 9 to 34 out of 1000 people are either completely deaf or have an acute hearing loss problem. Another major finding of the survey carried out is that deafness affects at least 1 person out of a 1000 before the age of 18 years. These statistics make it mandatory for the technological avant-garde to address the issue of bridging the communication gap between the hearing community and the deaf and hard of hearing community.

Ongoing research carried out over the past several years to address this issue has resulted in the development of new and sophisticated approaches to address this issue [2] [3] [4]. Various other approaches for the ASL finger-spelling have been suggested such as data gloves, neural networks, sensors. However, the Image processing technique is considered as a better approach because of various advantages such as:

- It is a more natural approach carried out under normal circumstances.
- No part of the process interferes with the signer.
- There is no distraction to the audience.
- Image compression techniques are available for data reduction which enhances the data processing as well as data storage.
- Feature extraction techniques are also available for fast processing as well as for reliability.

The feasibility of this approach is made possible due to the relative ease in implementing the data processing techniques and the large memory space available for the storage of data. This research strives to achieve the goal of bridging the communication gap between the deaf and hearing communities by creating a translation system which converts the American Sign Language

finger-spelling to digital English text. The present study was conducted to support the ongoing research ***project Sign2*** under *The Laboratory of Advanced Communication Technology* and *ECTET department*. The scope of this work is limited to ASL finger-spelling and does not include real-time processing of the data. However later phases of the ongoing Sign2 project will incorporate the real-time processing techniques of the data to attempt a successful conversion of the American Sign Language to digital text. The same approach can be applied for the various other sign language across different countries.

### 1.1 American Sign Language

Sign language is the primary means of communication with and among the deaf community. Baker et. al [5], have defined it as *Visual-gestural* language owing to the fact that the majority of sign languages incorporate the body rather than being restricted to just hand movements. However, as with spoken or any other type of language, there exist many forms and versions of the language. For clarity and efficiency, the ***Sign2 Project*** focuses exclusively on the American Sign Language, more commonly referred to as ASL. ASL is the official sign language of the United States and Canada[5] and is used by more than 500,000 people across the US and Canada. The history of the ASL dates back to year 1817, but it has developed over the period of time and is the fourth most commonly used language in the country [6].

One of the significant features of the language is its structure, which is devised keeping in mind the anatomy of human eyes. It is to be noted that ASL has one handed alphabets. Extensive research based on linguistic principles shows that American Sign Language consists of 18-19 hand shapes, 24 movements and 12 locations. Changing one of the parameters will change the meaning of the sign in a drastic fashion. Additionally, ASL has its own set of rules for the *creation of words* and *hand shapes*. However, ASL grammatically differs completely from spoken English but is comparable it in its expansion, breadth and depth. Like spoken English, ASL has many variations caused by difference in race, ethnicity, gender and age. In spite of these variations, ASL maintains a well-defined structure and grammar which makes it a complete language. The

three basic sentence structures [7] defined in ASL have been enumerated below

1. **Question** type sentences can be recognized by various gestures such as

   - Raised eyebrows
   - Slightly widened eyes
   - Forward tilt of head and / or body
   - Raised Shoulders
   - Simultaneous forward shift of body

2. **Command** can be identified in two different ways

   - To sign faster than usual
   - To sign much slower than usual

   both of these ways emphasize subject matter.

3. **Declarative sentences** When there is no grammatical signal, it means it is a declarative sentence.

### 1.2 Objective

While the final objective of the Sign2 Project is to enable real-time translation of standard conversational ASL, the scope of this work is contained in the objective of the first phase of the project. This research is carried out to bridge the communication gap between the deaf and hearing communities with the help of upcoming technological endeavors. Although real time translation is not within the objectives of this work, the study seeks to answer the following problems to justify the implementation of the approach undertaken.

- To devise a simple technique for the conversion of ASL finger-spelling into digital text.
- To acquire reliable and accurate results to validate the approach.
- To conducting the study in a natural environment with minimal or no artificial setup.

- To make the conversion as unobtrusive as possible for both the signer and the audience
- To employ a systematic implementation of the proposed methodology in future telecommunication technologies.
- To upgrade the methodologies with the use of existing systems and processes.

### 1.3 Background

As stated earlier various approaches have been proposed for the conversion of ASL-finger spelling in to text or audio. One of the popular approaches is the **device-based approach** where gloves, sensors etc were used as a measuring devices to analyze the hand movements. Waleed Kadous [2] in his work describes the understanding of 23 degrees of freedom along with 6 degree of freedom for 3D movements as the essential part of the *devise based approach*. On the other hand another popular approach is the **vision-based approach** wherein cameras are utilized to register the hand movements.

Glove based system suffers from the limitation of using a device which is intrusive both for signer and the audience. In the glove based system, finger flex sensors, tactile sensors and wrist positioning sensors are used to register the movements of the hand. Customizing gloves for varied sizes and hand shapes has proved to be a hindrance while employing the glove-based technology. To get accurate information 6D trackers are implanted, which not only makes the device bulkier but also expensive. Additional filtering becomes essential while analyzing the data due to the noise created by the trackers and sensors.

The vision-based approach uses cameras to capture the hand movements and in certain cases, more than one camera is employed to capture the 3D movement of hands. In previous studies, researchers have used sensor device places at the tip of the hand to measure accurately the position of the hand. In some of the approaches finger tips were color coded to differentiate the positioning and movement of hands. Though these techniques provided a

rigorous approach to the problem, they required more space for data storage and a large amount of processing power.

Another approach which was considered a feasible and competitive option was the **non device-based and non vision-based technology**. This technique is based on the theory of neural networks and is implemented using recurrent neural networks as specified in the work of Murakami et. al [3]. This system is built by registering occurrences and building a database based on a learning algorithm. But the construction of this database tends to be cumbersome and results in large processing times.

### 1.4 Thesis Road map

In this chapter the basic information useful in understanding the scope of thesis has been highlighted. Related work in the area of Sign Language conversion has been discussed in this section. Remainder of thesis is organized as follows:

**Chapter 2** highlights the approach to the problems discussed in the overview section. It also includes various subsections to the approach being considered.

**Chapter 3** subsequently discusses the results obtained using the approach discussed in the previous chapter.

**Chapter 4** enlists the long term goals of this study. It also discusses how this approach can be accommodated in the future technological advancements in the area of telecommunication.

**Chapter 5** summarizes the conclusions of the study carried out and the goals achieved so far.

To help the readers understand and follow the study, additional material(MATLAB code) has been included in the **Appendix** section.

# Chapter 2
# Approach

The approach [8] taken to cater in study is formulated into three steps

- Establishing a standardized set of physical measurements for ASL finger spelling.
- Generalize the measurements for different statistical range of subjects.
- Correlation of measurements with statistical range of subjects for letter recognition.

Image processing is used as a tool for the conversion of ASL into digital text in this particular investigation. It provides many advantages as compared to other techniques as it can be carried out in a more natural way and allows an additional advantage of presenting an unobtrusive and convenient solution for the signer as well as for the viewer. Furthermore, only one camera is required and hence this technology can be accommodated in telecommunication devices such as mobile phones and PDA's. One of the main advantages of the method is that since it employs image processing, it does not include complex calculations which consequently results in less processing overhead.

This study deals only with ASL finger-spelling and hence whole body movements are not considered. At this point real time processing of data is not considered, although, this aspect will be discussed in the later phases of the ***Sign2 project***. Figure 1 shows the block diagram for the approach as stated above.

## 2.1 Data Collection: Video Capture and Video storage

In the data collection phase of the project, videos of various subjects spelling out the alphabet in ASL were captured. Figure 2 shows the pictorial representation of the assembly used for capturing the video. Each of the subjects were made to spell out the complete alphabet. This process was carried out multiple times to capture many videos, so as to build an extensive database. The initial

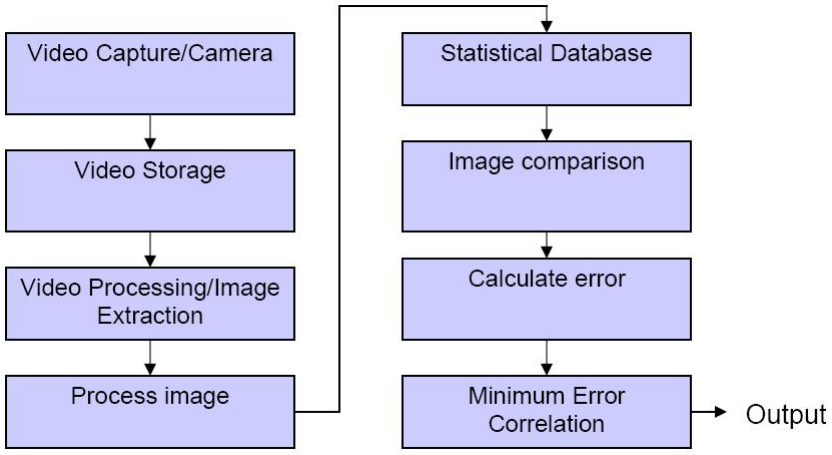


Figure 1: The Sign2 System functional block diagram

phase of data collection included videos which were captured in an uncontrolled environment. Certain abnormalities like reflective effects from regions of the hand were observed. Refer to the Appendix A for the cautions and methods to be followed while capturing videos for populating the database. In the later phase, videos for the database were taken in a more controlled environment. Videos were captured using the facilities of Graphic Design Lab in the College of Applied Science and Technology. **Quick Time** was used to convert the captured images into AVI format. AVI format is mandatory because MATLAB only recognizes videos in AVI format. The videos captured were of the size 720 x 480 pixels. To reduce the size to 320 x 240 and to uncompress the Videos, Fx Video Converter Demo 6.3.0 was used.

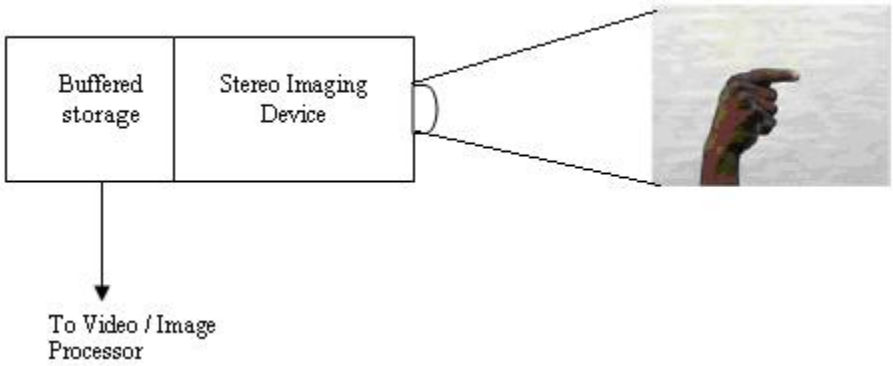


Figure 2: Imaging system Illustration

## 2.2 Data Processing

Data Processing consists of three phases

1. Video Processing and Image Extraction
2. Image Processing
3. Image Storage

### 2.2.1 Video Processing and Image Extraction

This phase of the project consists of post processing of the captured video. The captured video is read in MATLAB. The system allows the flexibility to specify the start and end frame of the video for which the processing is carried out. Individual frames are extracted as images which are used in the image processing phase of the project. Image extraction is not only used for the statistical database but also for the extraction of images from the test video for further comparison with the already built statistical database. Figure 3 shows the extraction of a frame from a video.

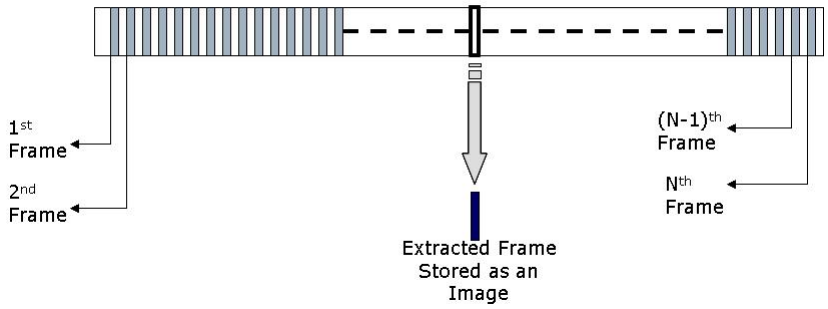


Figure 3: Frame Extraction from a Video clip

### 2.2.2 Image Processing

Image processing is used to process the extracted images. Frames extracted by the image extraction technique are converted to black and white images pixel by pixel. Images obtained from image extraction are color images, the red, green and blue (RGB) components of the image are extracted separately. Each component is converted to double precision and binary thresholding is performed based on the user specified threshold value. Thus, image gets

converted into black and white. This black and white image is cropped based on the detection of the edges (left, right and top edges) of the hand. Because of the varying hand sizes of the subjects, the resulting cropped images are of different sizes. To counter this problem and to provide a generalized system, the cropped images of all subjects were again resizes to a consistent size of 150 x 80.

The following Figure 4 shows the graphical user interface developed to post process the captured video in order to extract the frames from it. This GUI has 3 main modules : first module is *display module* to show the post-processing of the video, second module shows the black and white cropped image of the last frame extracted and third module shows the error graph between the extracted frames. This GUI allows a user to specify the frame difference (module 6) between the two consecutive frames extracted in the text box specified as the Frame Difference. Module 5 specifies the start as well as the end of frame of the video. In the example shown in the figure, every third frame will be extracted and processed as defined in the frame difference module. The graph shown in the GUI depicts the error between the original frames extracted as well as the processed frame.

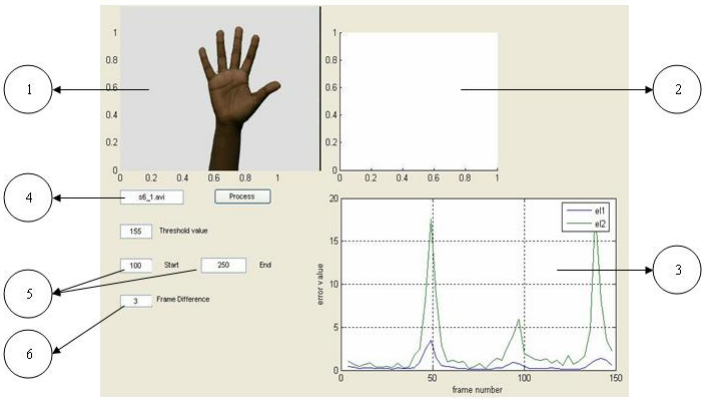


Figure 4: Frame Extraction GUI

### 2.2.3 Image Storage (Statistical Database)

The heart of the study lies in the statistical database, which determines the reliability of the results. It becomes crucial to capture as many hand structures

as possible for the generalization of this methodology. To improve the reliability, each subject was asked to sign the alphabet from A to Y repetitively. The repetitive technique is used to record the changes in the hand gestures of the signs at different instances. The database consists of the black and white resized images extracted using the image processing and cropping techniques as defined in the image processing subsection. To populate and define this database, a naming convention is necessary. Therefore the study has given each subject a corresponding identification. For instance S1_A2, where S stands for the subject followed by a number ranging from 1 to $n$[1]. A stands for a letter from A to Y followed by another number which represents the number of iterations taken while capturing the video.

### 2.3 Image Comparison and Letter Recognition

This is the most exciting part of the study as it involves the letter recognition. Let's assume that we are working with a test image extracted from a video captured in the data collection phase. This image is again cropped using the image processing technique described in the image processing section. Now the cropped and resized test image is compared with all the images in the statistical database images based on the definition of **Mean Square Error(MSE)**and **Peak signal to Noise Ratio(PSNR)**. As explained above, phases of image comparison are shown in the Figure 5.

$$MSE = \frac{1}{LW} \sum_{l=1}^{L} \sum_{w=1}^{W} (I(l,w) - I'(l,w))^2. \tag{1}$$

where, I is the original image and I'is the new decompressed image.

$$PSNR = 20log_{10} \left[ \frac{255}{\sqrt{MSE}} \right] \tag{2}$$

With this definition in mind, an error matrix is created having the error values for corresponding image in the database. The set of image in the database that corresponds to a given letter and has the lowest cumulative error, revels the highest priority of the correct letter being returned which is clearly shown in

[1]n is a natural number

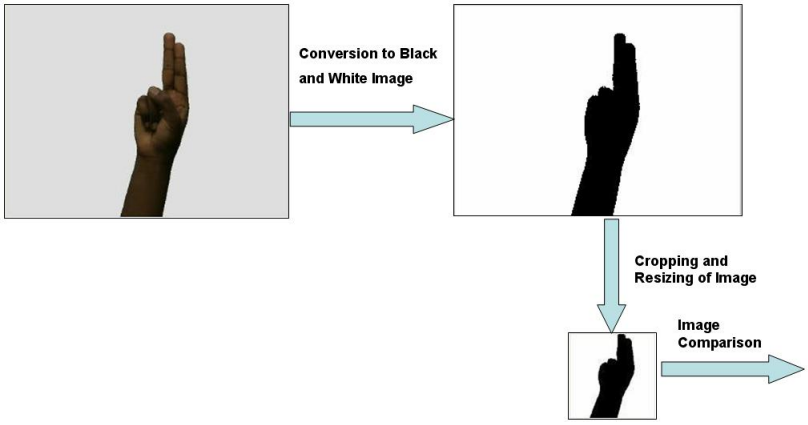


Figure 5: Phases of Image Comparison

Figure 6.

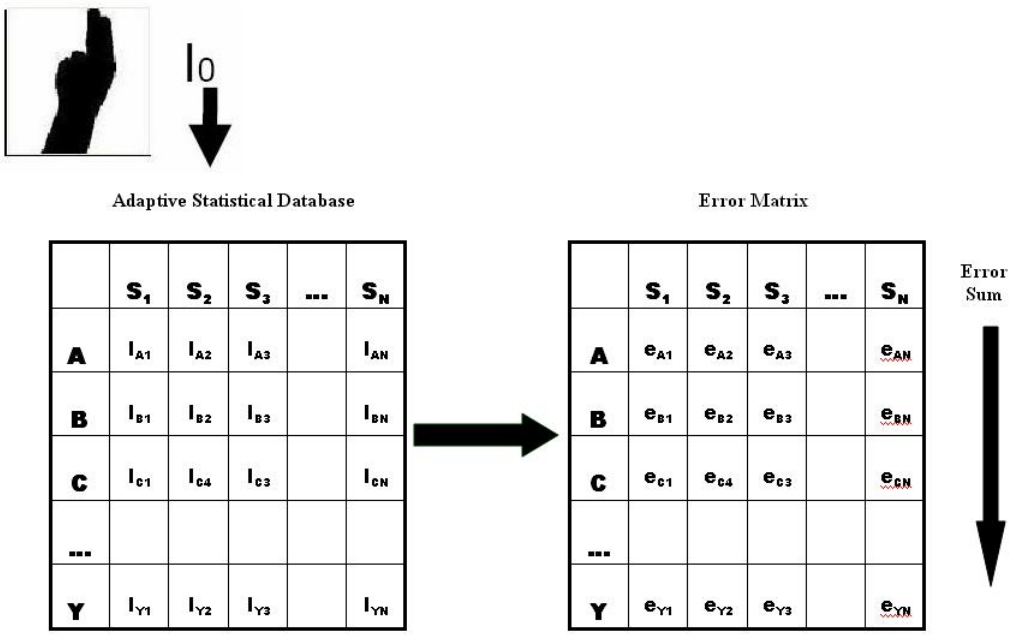


Figure 6: Statistical database and resulting Error Matrix

Similar to the Graphical user interface generated for frame extraction in the Image processing section, an Alpha_Sin2 GUI is developed as shown in the Figure 7 . The various sections of the GUI are listed below:

- Display module shows the post processing of the videos.
- There is a Module for displaying the error graph between the consecutive

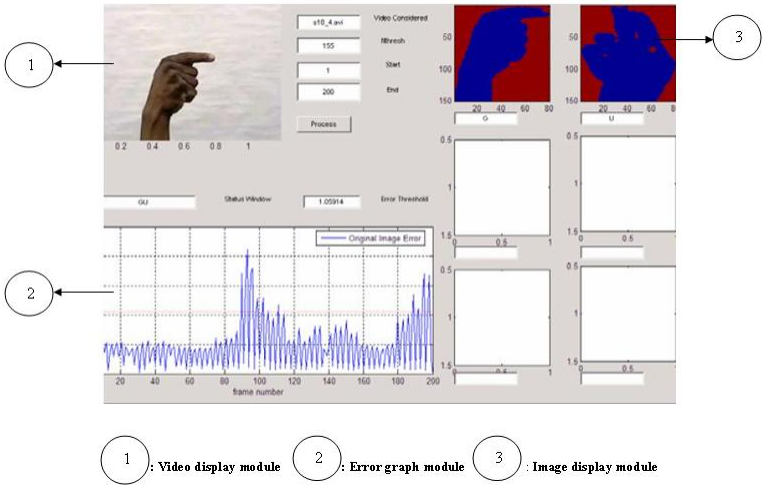


Figure 7: Graphical user interface

frames.

- Six modules are used to display the frames extracted and populated with the letter being recognized.

- Processing push-button initiated the processing based on the filed values specified in the text boxes, Video considered, threshold value (Nthresh), starting of the video (Start) and end of the frame(End).

To further understand the letter recognition phase of this investigation, let us consider an example. A video is considered where the word $LAY$ is being spelled. The video is run at the rate of 30 frames per second and 399 frames are being considered. Hence total duration of the video is approximately 13.5 second.

As per the assumption, when a person is signing he will hold the letter for some time so that the other person can register it. During that time there will be very little error difference between the consecutive frames as there is no hand movement involved. This marks the presence of a letter being spelled. The sudden change in the error between the consecutive frame shows the transition from one letter to an other. Figure 8 indicates the error between the consecutive processed frames. Moreover, the error between the consecutive frames should be constant for at least 10 to 20 frames before considering the letter actually

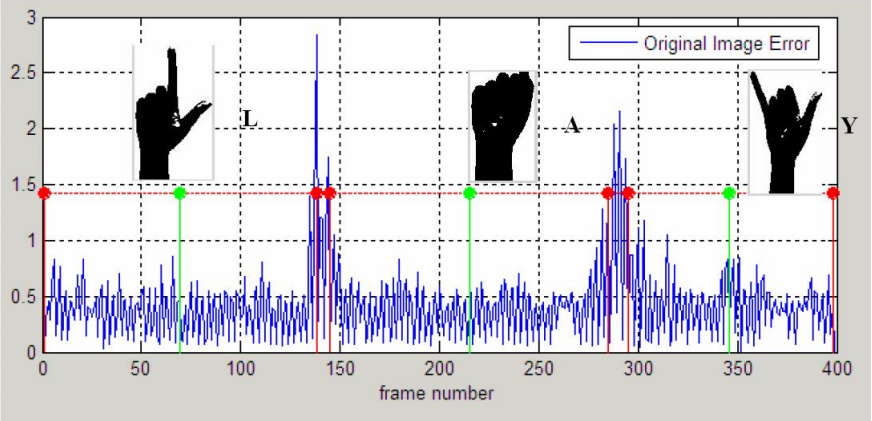


Figure 8: Frame to Frame error calculation and letter recognition

being spelled. An error threshold ($e_{thresh}$) is calculate according to the formula shown below.

$$e_{thresh} = \frac{3}{10}(e_{max} - e_{min}) \tag{3}$$

Where $e_{max}$ is the maximum error and $e_{min}$ is the minimum error between the frames.

As per the assumption, for this particular example $e_{thresh}$ is calculated as 1.4178, $e_{max}$ is 2.8356 and $e_{min}$ is 0. In this particular case error is constantly below the threshold for frame 1 to 138. Therefore, frame 69 is extracted which represents the mid frame of this frame stack. Similarly frame 215 and 346 are selected for the next two letter recognition from the frame stack from 145 through 285 and from 295 through 398. These extracted frames are again processed and compared with the database images in the same fashion as described in the image processing and image comparison and letter recognition section. There are frames for which the error is less than $e_{thresh}$ but never remains constant for 10 consecutive frames and are discarded.

# Chapter 3

# Results

By undertaking the approach defined in the previous chapter, results are recorded based on the data gathered by capturing the videos of different subjects spelling out English alphabet and words. The videos consist of letters from A to Y and three, four or five letter words specified in the following list.

| **Three Letter Word** | **Four Letter Word** | **Five Letter Word** |
|---|---|---|
| ASL | CARD | THINK |
| HAT | WORD | SHARE |
| LET | DEAF | PAPER |
| LAY | BABY | PRINT |

Table 1: List of the words considered

### 3.1 Phase I: Alphabet Recognition

First phase of the analysis is based on the calculation of Recognition Ratio ($\gamma$) of English alphabet. Whenever a letter is recognized correctly, it is given a value 1 for each iteration and if it is not recognized correctly it is assigned a value 0. This concept is represented in Table 2.

Hence, Recognition Ratio ($\gamma$) for an alphabet is defined as

$$\gamma = \frac{\sum_{i=1}^{n}(\alpha_i)}{n} \tag{4}$$

Where, $\alpha$ is the total number of successful recognitions of a letter and it ranges from $A_1$ to $A_{25}$.

To further understand this concept, Let's take a case of letter I, if *total number of iterations (n)* is 15 and *total number of successful recognition ($\alpha$)* is 14. Therefore,

$$\gamma = \frac{14}{15}$$

| **Alphabet** | **Iteration 1** | **Iteration 2** | **Iteration 3** | ... ... | **Iteration n** | **Recognition Ratio ($\gamma$)** |
|---|---|---|---|---|---|---|
| $A_1$ | 1 | 0 | 1 | ... | 0 | $\frac{1}{n} * \sum_{i=1}^{n} (A_1)_i$ |
| $A_2$ | 0 | 1 | 1 | ... | 1 | $\frac{1}{n} * \sum_{i=1}^{n} (A_2)_i$ |
| $A_3$ | 1 | 1 | 1 | ... | 0 | $\frac{1}{n} * \sum_{i=1}^{n} (A_3)_i$ |
| ... ... | | | | | | |
| $A_{25}$ | 0 | 0 | 1 | ... | 1 | $\frac{1}{n} * \sum_{i=1}^{n} (A_{25})_i$ |

Table 2: Recognition Ratio Calculation for Alphabet

Hence,

$$\gamma = 0.93333$$

And,

$$\gamma\% = 93.333$$

Table 3 shows the various results recorded for letter A to Y,
Figure 9 shows the graphical representation of Recognition Percentage against each alphabet,
Graph shown in the Figure 9 shows that letters have different degrees of recognition percentages ($\gamma\%$). Letter $L$, $V$ and $Y$ have highest recognition percentage of 100% whereas letter $R$ has lowest $\gamma\%$ of 37.5.

### 3.2 Phase II: Word Recognition

As our analysis also deals with the recognition of words, Estimated Recognition Ratio $\omega$ for a Word is defined as

$$\omega = \frac{\sum_{j=1}^{c} (\gamma_j)}{c} \tag{5}$$

Where, c is the number of letters in a word.

| **Alphabets** | **Total Number of Iterations** | **Successful Recognition** | **Recognition Ratio** $\gamma$ | **Recognition Percentage** $\gamma\%$ |
|---|---|---|---|---|
| $A$ | 14 | 11 | 0.78571 | 78.571 |
| $B$ | 12 | 11 | 0.91666 | 91.666 |
| $C$ | 15 | 6 | 0.40 | 40 |
| $D$ | 12 | 11 | 0.91666 | 91.666 |
| $E$ | 15 | 11 | 0.73333 | 73.333 |
| $F$ | 11 | 7 | 0.63636 | 63.636 |
| $G$ | 14 | 13 | 0.92857 | 92.857 |
| $H$ | 14 | 13 | 0.92857 | 92.57 |
| $I$ | 15 | 14 | 0.93333 | 93.333 |
| $J$ | 11 | 8 | 0.72727 | 72.727 |
| $K$ | 10 | 8 | 0.80 | 80 |
| $L$ | 10 | 10 | 1 | 100 |
| $M$ | 13 | 6 | 0.46153 | 46.153 |
| $N$ | 13 | 7 | 0.53846 | 53.846 |
| $O$ | 13 | 8 | 0.61538 | 61.538 |
| $P$ | 11 | 8 | 0.72727 | 72.727 |
| $Q$ | 13 | 12 | 0.92307 | 92.307 |
| $R$ | 8 | 3 | 0.375 | 37.5 |
| $S$ | 11 | 6 | 0.54545 | 54.545 |
| $T$ | 9 | 4 | 0.44444 | 44.444 |
| $U$ | 5 | 6 | .80 | 80 |
| $V$ | 9 | 9 | 1 | 100 |
| $W$ | 7 | 6 | 0.85714 | 85.714 |
| $X$ | 4 | 3 | 0.75 | 75 |
| $Y$ | 8 | 8 | 1 | 100 |

Table 3: Reliability Ratio for Alphabet

Alternatively $\omega$ can also be expressed in the following manner,

$$\omega = \frac{\sum_{j=1}^{c}\sum_{i=1}^{n}(\alpha_{ji})}{nc} \tag{6}$$

Lets take a case where estimated recognition ratio for word $ASL$ is to be calculated, Therefore,

$$\omega_{asl} = \frac{\gamma_a + \gamma_s + \gamma_l}{3}$$

By refereeing to table 3, $\gamma_a$ is 0.78571, $\gamma_s$ is 0.54545 and $\gamma_l$ is 1 , therefore Estimated Recognition Ratio for word ASL ($\omega_{asl}$) is 0.7770.

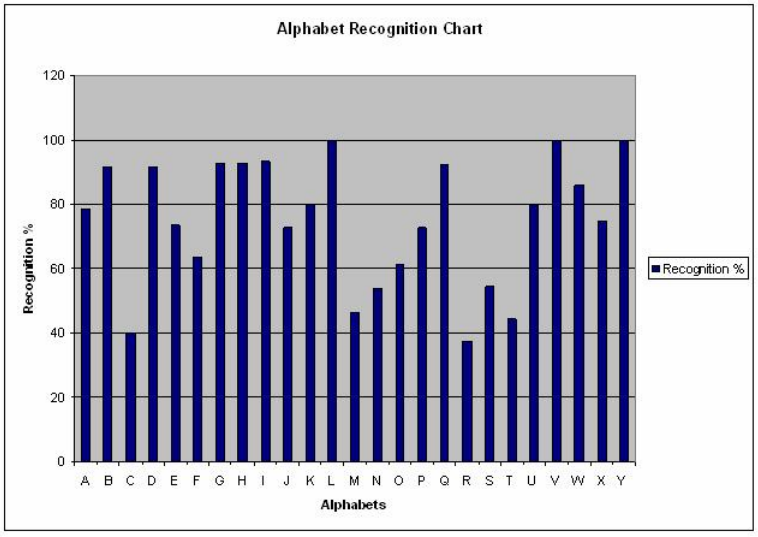


Figure 9: Alphabet Recognition Percentage Chart

Further, for Experimental Recognition Ratio Calculation for word, Weight Ratio Assignment is used. Weight Ratio for word ($Wt_r$) can be calculated based on the following formula,

$$Wt_r = \frac{N_{cr}}{c} \tag{7}$$

Where, $N_{cr}$ is the Number of Correctly recognized letters in a word and $c$ is the number of letters in a word.

Various Weight Ratios are tabulated for three letter, four letter and five letter words are shown in the table 4,

Again, consider the word $ASL$ for the experimental Recognition Ratio Calculation ($\omega_{exp}$). If $ASL$ is recognized as $AAL$ in first iteration, $ASL$ and $ABC$ in second and third iteration respectively then based on the weight assignment $\omega_{exp}$ can be defined in the following manner,

$$\omega_{exp} = \frac{Weight_1 + Weight_2 + ..... + Weight_i}{N_i} \tag{8}$$

| Words | Successful Recognition | Recognition with one error | Recognition with two error | Recognition with three error | Recognition with four error | Recognition with five error |
|---|---|---|---|---|---|---|
| 3 Letter Word | 1 | 2/3 | 1/3 | 0 | ... | ... |
| 4 Letter Word | 1 | 3/4 | 2/4 | 3/4 | 0 | ... |
| 5 Letter Word | 1 | 4/5 | 3/5 | 2/5 | 1/5 | 0 |

Table 4: Reliability Ratio for Words

where, $N_i$ is Number of Iterations performed.

In this case $Weight_1$ comes out to be 2/3 as only 2 of 3 letters were recognized correctly. Similarly, $Weight_2$ and $Weight_3$ comes out to be 1 and 1/3 respectively. As number of Iterations under consideration are three, therefore $N_i$ is 3. Therefore, Experimental Recognition Ratio($\omega_{exp}$) for word **ASL** comes out to be 0.66667 and the Experimental Recognition Percentage ($\omega_{exp}\%$) is 66.667. Table 5 shows the experimental recognition ratio, experimental recognition percentage and estimated recognition percentage for Words under consideration as tabulated in table 1,

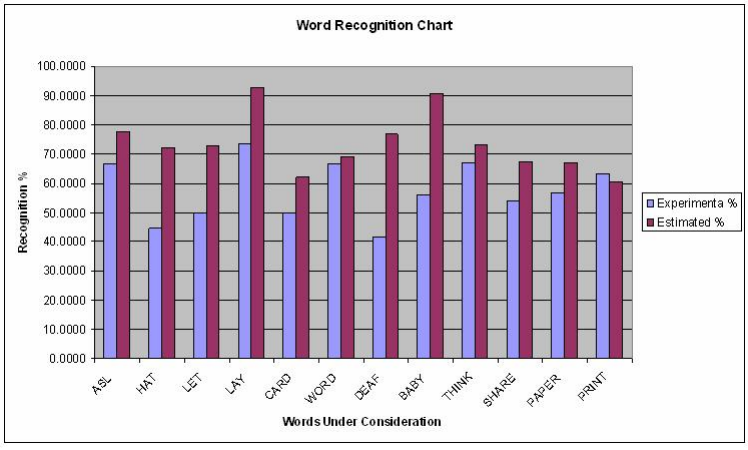


Figure 10: Word Recognition Chart

Figure 10 presents contrast between experimental recognition percentage and estimated recognition percentage for each word under consideration. For

| **Word** | **Experimental Recognition Ratio** $\omega_{exp}$ | **Experimental Recognition Percentage** $\omega_{exp}\%$ | **Estimated Recognition Ratio** $\omega$ | **Estimated Recognition Percentage** $\omega\%$ |
|---|---|---|---|---|
| $ASL$ | 0.66666 | 66.666 | 0.77705 | 77.705 |
| $HAT$ | 0.44444 | 44.444 | 0.71957 | 71.957 |
| $LET$ | 0.49999 | 49.999 | 0.72592 | 72.592 |
| $LAY$ | 0.73333 | 73.333 | 0.92957 | 92.957 |
| $CARD$ | 0.50000 | 50 | 0.619345 | 61.9345 |
| $WORD$ | 0.66667 | 66.667 | 0.69104 | 69.104 |
| $DEAF$ | 0.41667 | 41.667 | 0.76801 | 76.801 |
| $BABY$ | 0.5625 | 56.25 | 0.90476 | 90.476 |
| $THINK$ | 0.67142 | 67.142 | 0.72896 | 72.896 |
| $SHARE$ | 0.54000 | 54 | 0.673614 | 67.3614 |
| $PAPER$ | 0.56667 | 56.667 | 0.66971 | 66.971 |
| $PRINT$ | 0.63333 | 63.333 | 0.60370 | 60.370 |

Table 5: Estimated and Experimental Reliability Ratio for Words

the word *WORD*, the difference between estimated recognition ratio and experimental recognition ratio is the lowest (2.437%), whereas this difference is highest for word *DEAF* (35.134%).

# Chapter 4
# Future work

With regards to telecommunication, this technology will enable more free form and initiate communications by allowing the deaf community to bypass the limits of standard text entry similar to the ways voice recognition software allows for natural dictation for the hearing community. As telecommunication devices increasingly move towards hybrid or *convergent* designs, Voice will increasingly become the primary method for interacting with these devices. This can already be seen in the Voice dialing capabilities that are already virtual standard on new mobile phones. With the ability to recognize sign languages especially natural forms, the deaf community will not be left as the industry moves away from the standard text inputs. Additionally, with the exceptional increase in the power, performance and the capability of the electronics that telecommunications devices operate on, the development of a software based sign language interpreter may usher in a new era of the use of sign language by hearing people. As the interpreter becomes more accurate, it should offer a very attractive alternative to the current very popular use of Repetitive Stress Injury (RSI) inducing text message on alpha numeric keypads.

Currently, text messaging is most often used in situations where voice conversations are not wise or out-right prohibited. Such prohibitions are increasing with the increase in subscribers. Sign language allows the same level of fluidity, intimacy and range of expressions as the actual voice conversation without the disruptive noise. Thus it has significant advantage over the former, trying to express oneself using little keys.

The immediate application of this technology is to bridge the communication gap between deaf/hard of hearing and hearing communities. This application could be either implemented as a full fledged software application or the application could be embedded on the chip using ASIC design techniques to facilitate the communication gap between hearing impaired and hearing communities. One of the major advantages of using image processing as a tool is that the

data processing is faster. This can further be increased by using feature extraction methods [9] which reduces the amount of data to be processed. While developing this system as a full fledge software or hardware application the need will arise to keep the statistical database size on the disk to be minimum. This can be achieved by numerous compression techniques available [10].

Once the system is implemented on a chip, mobile phones and PDA's with a camera could be used in an effective manner by the Deaf/Hard of hearing community. They could send messages by sign language which they are already aware of rather than typing text. One of the other applications of this system could be in emergency systems for the hearing impaired whereby the system could send messages by converting and recognizing the sign language and sending it in the form of text/voice to the concerned authorities.

# Chapter 5
# Conclusion

The Primary focus of this study was to examine image processing as a tool for the conversion of American Sign Language in to digital text. Further this study promises to be used in the real time application to fully recognize American Sign Language. This can be further developed into a system which can be integrated in to the upcoming telecommunication devices with cameras to bridge the communication gap between the hearing and deaf/hard of hearing communities. System can be enhanced in terms of increase in the data processing speed and data storage by using the compression techniques and feature extraction techniques.

The results observed after conducting the experiments strengthens the approach being considered as an effective means to tackle the problem. Discrepancies observed such as faulty recognition of letter or inability to register the letters while gathering results is due to the assumptions under consideration such as the window size calculated for letter recognition and the error threshold calculation. Reliability in recognition of the correct word can be increased if we devise an approach to identify the pace of signing of each subject. This would increase the probability of identifying the frame containing the correct letter. Further, by implementing the background reduction and Shadow reduction techniques improves the results.

**List of References**


[1] Deaf and H. of Hearing Services @ Texas, "Hearing loss statistics and demographics," 2005.

[2] M. Kadous, "Grasp: Recognition of australian sign language using instrumented gloves," 1995.

[3] D. Yarowsky, "Gesture recognition using recurrent neural networks," pp. 237–242, Journal of the ACM, January 1991.

[4] K. Alkoby and E. Sedwick, "Using a computer to fingerspell," *DeafExpo 99, San Diego, CA*, November 1999.

[5] C. Baker-Shenk and D. Cokely, *American Sign Language: A Teacher's Resource Text on Grammar and Culture*. Washington, D.C.: Clerc Books, Gallaudet University Press, 1980.

[6] D. Burgett, "Assistive technology: Communication devices for the deaf," 2004.

[7] S. M. Inc., "Introduction to american sign language," 2002.

[8] C. M. Glenn, D. Mandloi, K. Sarella, and M. Lonon, "An image processing technique for the translation of asl finger-spelling to digital audio or text," NTID International Symposium Instructional Technology and Education of the Deaf, June 2005.

[9] K. Fukunaga and W. L. G. Koontz, "Application of the karhunen-loeve expansion to feature selection and ordering," *IEEE Trans. Comput.*, vol. C-19, no. 4, pp. 311–318, April 1970.

[10] C. M. Glenn, M. Eastman, and G. Paliwal, "A new digital image compression algorithm base on nonlinear dynamical system," IADAT International Conference on Multimedia, Image Processing and Computer Vision, Conference Proceedings, March 2005.

# Chapter A
# Video Capturing And Recording

Certain observations were made in the process of capturing videos for a model database. The process of capturing the video demands a certain method and caution.

1. The video should be captured in moderate and even lighting. Lighting is an important factor to be considered while capturing the video. The lighting should not be either too bright or too dim. Bright light produces a reflective effect from regions of the hand like the lighter part of the hand (i.e., the palm) and the level surfaces of the hand. This results in corrupting the black-and-white version of the letter frame and recording inaccurate error values. Daytime is the most appropriate time to record the videos, if the recording is being done in the laboratory without any peripheral lighting equipment as, there is an even distribution of light as opposed to recording during the nights when we have to depend solely upon the lighting arrangement of the laboratory.

2. The ideal distance from which the video ought to be shot should be around one and half to two feet. Caution should be exercised not to place the subject too far away from the capturing device as this would result in improper and erroneous results. Placing the subject too near would result undesirable cropping of the subject while changing from one letter to another during finger spelling, once again resulting in erroneous results.

3.The background used behind the subject while capturing video should be even and without creases. Caution should be exercised to ensure that the background behind the subject is even and clear of creases, as this will produce an uneven reflective effect from the background and will consequently corrupt the database.

4.Avoid hand and finger accessories of any sort while finger spelling. It is important for the subject not to accessorize the finger spelling hand during the recording process. Any accessories worn by the subject would result in undesirable and uneven reflection of light. This interferes with the black-and-white version of the letter frame which is to be stored in the database producing and recording inexact error values.

# Chapter B
# Error Thresholding

```
function [Errorthresh, arr, error_arr, count2,
         startarr, endarr, midarr] = error_thresh_1(M1,Nthresh,
                                filename,fstart,fend, arr_images)
dframe = 1;
fframe2 = 1;
flag1 = 0;
arr(1:fend)=0;
size(arr);

%ERROR ARRAY CALCULATION

 for b=fstart:(fend-1)
    fframe1 = fframe2;
    fframe2 = fframe2 + dframe;
    [IME1,Map] = frame2im(M1(fframe1));
    [IME2,Map] = frame2im(M1(fframe2));

% CALCULATE ERROR

    eIM1 = IMAGE_ERROR(IME1,IME2);
    arr(b) = [eIM1];
 end

%ERRORTHRESHOLD CALCULATION
    minError = min(arr);
    maxError = max(arr);
    Errorthresh = (maxError - minError)/2;
    count = 0;
    count2 = 1;
```

```
    for a=fstart:(fend-1)
        if (arr(a) < Errorthresh & a ~= fend - 1)
            count = count+1;
            flag1 = 1;
        elseif (flag1 == 1 & count >= 20)
%WINDOW SIZE CALCULATION
        startarr(count2) = (a - count);  %window Start array
        count = round(count/2);
        count = a - count;
        midarr(count2) = count;          %Mid Frame of the window
        endarr(count2) = a;              %window End array
        [IME3,Map] = frame2im(M1(count));
        ITHR1 = IMAGE_PROCESS(IME3,Nthresh);

% RESIZING AND PLOTTING OF THE FRAME
        ITHRE1 = imresize(ITHR1,[150 80]);
        convertIME = uint8(ITHRE1);
        subplot(arr_images(count2));
        imagesc(convertIME);
        fstar = num2str(count);
        errframe = ['Frame\' fstar '.jpg'];
        errframe = cellstr(errframe);
        error_arr(count2) = errframe
        count2 = count2+1;
        errframe = char(errframe);
        imwrite(convertIME, errframe);
        count = 0;
        flag1 = 0;
        else
            count = 0;
        end
    end
```

# Chapter C

# Image Processing

```
function ITHRESHP = IMAGE_PROCESS(IM,Nthresh)
% 4.0 PROCESSING
IMred = IM(:,:,1);
IMgreen = IM(:,:,2);
IMblue = IM(:,:,3);
%   4.1 CONVERT TO DOUBLE PRECISION
IMredD = double(IMred);
IMgreenD = double(IMgreen);
IMblueD = double(IMblue);
IMsumD = (IMredD+IMgreenD+IMblueD)/3;
IMsum = uint8(IMsumD);
%   4.2 BINARY THRESHOLDING
ITHRESH = 255*(IMsumD>=Nthresh);
ITHRESH(:,1:60) = 255;
NV = length(ITHRESH(:,1));
NH = length(ITHRESH(1,:));
%   4.3 FIND AND CROP EDGES
%       4.3.1 LEFT EDGE
for v = 1:NV
    h = 1;
    while (ITHRESH(v,h) == 255) & (h < NH)
        h = h + 1;
    end
    bp(v) = h-1;
end
bpleft = min(bp);
ITHRESHP = ITHRESH(:,bpleft:NH);
NHP = length(ITHRESHP(1,:));
clear bp
```

```
%       4.3.2 TOP EDGE
fid = fopen('verical.txt','w');
for h = 1:NHP
    v = 1;
    while (ITHRESH(v,h) == 255) & (v < NV)
        v = v +1;
    end
    bp(h) = v-1;
end
fprintf(fid,'bp(h) %d\n',bp);
fclose(fid);
bptop = min(bp);
ITHRESHP = ITHRESHP(bptop:NV,:);
%       4.3.3 RIGHT EDGE
clear bp
NVP = length(ITHRESHP(:,1));
for v = 1:NVP
    h = NHP;
    while (ITHRESHP(v,h) == 255) & (h > 1)
        h = h - 1;
    end
    bp(v) = h;
end
bpright = max(bp);
ITHRESHP = ITHRESHP(:,1:bpright);
```

# Chapter D
# Sign 2 GUI

```
%INITIALIZATION

global ITHRESHPR1
global M1
global Nthresh
global filename
global fstart
global fend
global x
global arr

set(handles.edit18,'String','Processing Video');
arr_handles = [handles.edit12 handles.edit13 handles.edit14
               handles.edit15 handles.edit16 handles.edit17];
arr_images = [handles.axes8 handles.axes9 handles.axes10
               handles.axes11 handles.axes12 handles.axes13];
crop = 150:80;
 for i = 1:6
     subplot(arr_images(i));
     imagesc(crop);
     set(arr_handles(i),'string',' ');
 end;
Nthresh = str2num(get(handles.edit6,'String'));
%OPENING A TEXT FILE WITH THE WRITE PERMISSION
fid = fopen('arr.txt','w');
%READ THE VIDEO FROM START TO END
filename = ['Video Samples\',get(handles.edit1,'String')];
fstart = str2num(get(handles.edit4,'String'));
```

```
fend = str2num(get(handles.edit5,'String'));
Nframes = fend-fstart+1;
M1 = aviread(filename,fstart:fend);
axes(handles.axes1);
movie(M1);
x = 1:fend;
%error_thresh FUNCTION CALL
[Errorthresh, arr, error_arr, count2, startarr, endarr, midarr]
= error_thresh_1(M1,Nthresh,filename,fstart,fend, arr_images);
set(handles.edit18,'String','Plotting Error Graph');
%PLOTTING GRAPH
subplot(handles.axes4)
plot(x,arr,x,Errorthresh,'-.r');
hold on
z = size(startarr);
y(z) = Errorthresh;
stem_handle1 = stem(startarr,y,'r','fill');
stem_handle2 = stem(endarr,y,'r','fill');
stem_handle3 = stem(midarr,y,'g','fill');
xlabel('frame number')
ylabel('error value')
legend('Original Image Error')
grid on
hold off

%Alpha_reco FUNCTION CALL

Ind = Alpha_reco(error_arr, count2, arr_handles);
set(handles.edit18,'String',Ind);
set(handles.edit11,'String',Errorthresh);
```