Contemporary Issues Summit, Harvard, Boston, USA, March 2017, Vol. 13, No. 1
ISSN: 2330-1236

Video-Based Finger Spelling Recognition for Ethiopian Sign Language Using Center of Mass and Finite State of Automata

Eyob Gebretinsae
Addis Ababa Institute of Technology, AAU
Center of Information Technology and scientific computing
Addis Ababa, Ethiopia

Abstract

In this paper we describe a method for automated recognition of Ethiopian Sign Language (ESL) finger spelling from a video. The method automatically selects images from a given movie. To select appropriate images for processing from a given movie we develop a computational algorithm and we test suitable values for the algorithm. To understand the meaning of the selected images, it applied image pre-processing techniques, global thresholding, grouping neighbourhood and calculating canter of mass on the selected images. After applying these techniques, the method uses finite state automata to recognize the ESL finger spellings. Besides of this the new recognition method select suitable processed image for consonant recognition. The method recognizes the seven vowels of ESL. The method is experimented using the 238 ESL finger spelling and achieved 91% recognition performance; through which each of the seven vowels have 34 representations from each of the 34 consonants. As a result, the method is appropriate to recognize the ESL finger spellings integrating with the previous or future works on ESL consonant recognition.

Keywords: Sign Language Recognition, Finite State Automata and Video Understanding.

Introduction

Sign languages are the basic means of communication between hearing impaired people. It is made up of an organized system of signs. This includes gestures, mimes and facial movements. Sign language is usually used by the deaf people, or the hearing people who can communicate with deaf people. According to Aleem, Yousuf, Mehmood, Suleman, Sameer, Razi, Rehman and Israr, 2005, sign language is not universal. It varies from country to country or regions within countries. Ethiopian Sign Language (ESL) is the sign language of the deaf in Ethiopia. ESL is complete in both signing and finger spelling. Signing includes the conceptual sign expressions which are dominantly applied to convey meaning in ESL (Masresha Tadesse, 2010) and finger spellings are alphabets used to spell scientific words and names. However, the Ethiopian Sign Language is not well studied and still it is on the infant stage of development. Therefore, a number of research works need to be done in order to address the special needs of the hearing-impaired community of Ethiopia.

Hearing-impaired people usually have communication problems when they want to communicate with hearing people without signing skill. A translator is usually needed when a deaf person wants to communicate with persons that do not speak sign language (Vassilia & Konstantinos, 2002). However, they cannot depend on interpreters every day in life mainly due to the high costs and the difficulty in finding and scheduling qualified interpreters (Aleem, et al., 2005). Therefore, this paper have a contribution in the study of Ethiopian sign language translator. The objective of the research is to develop a method for recognition of Ethiopian Sign Language finger spellings from a video. It introduces a new method on ESL finger spelling recognition.

Overview of Ethiopian sign language finger spelling

Ethiopian sign language has sign alphabets called Ethiopian manual hand alphabets (EMA) (National Interpreter Resource, 2010). These alphabets are used to spell scientific words, names and meanings which do not have single signed words. According to (Masresha tadesse, 2010), ESL has 33 manual alphabets with their corresponding seven movements for each of the 33 alphabets. In 2009, additional one signed alphabet was set for the Amharic letter Ve/ (Masresha tadesse, 2010). Unlike American Sign Language (ASL), ESL finger spellings have movements that change the meaning of the sign letter. According to (Ricco and Tomasi, 2009), ASL finger spellings do not require motions for most of the letters. Instead most of the letters are primary distinguished by the hand shape. Conversely, ESL finger spellings represent by hand shape and motion. ESL finger spelling represents Amharic consonant series with hand configurations, seven movements correspond to the seven Amharic vowel orders (Kyle Duarte, 2010) ( ). This makes Ethiopian sign language continuous. In continuous sign languages, motion detection is essential for recognition of the language. Fig. 1 shows the seven movements of ESL finger spellings that correspond to the seven Amharic vowel orders.

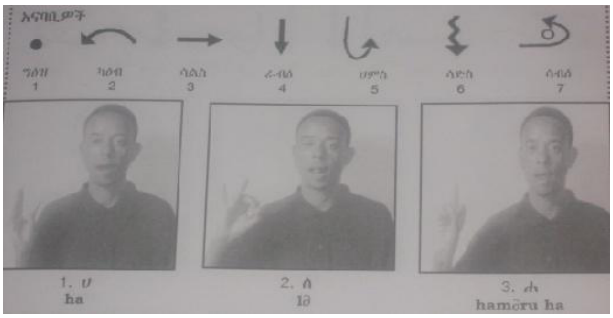


Figure 1 Samples of ESL finger spelling with their seven movements (Ethiopian National Association of the Deaf, 2008)

This study recognizes the seven vowels of each of the 34 Ethiopian sign language manual alphabets and it recommends which picture frame is used for the recognition of the consonant from set of frames. On this research we use center of mass to detect the motion and finite state automata to recognize the movements. In addition, global thresholding and other algorithms are also applied before recognition of the signs.

Signed Hand Segmentation and Feature Extraction

To segment the signed hand and to extract features, we use four basic steps. These are video acquisition, frame selection, signed hand segmentation, and feature extraction.

Video Acquisition

Captured video of Ethiopian sign language finger spellings is first acquired by the system. Figure 2 shows sample video input to the system.

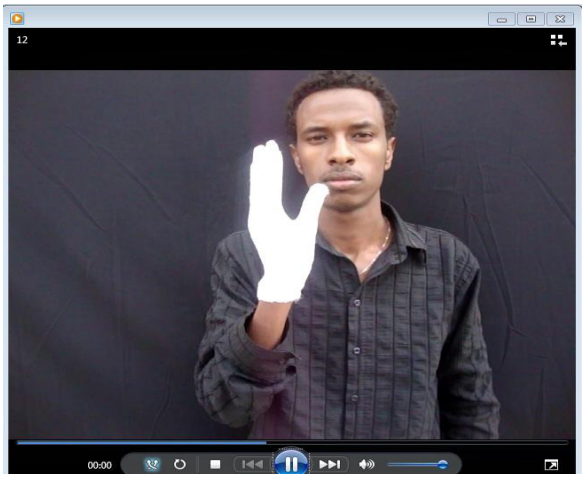


Figure 2 Sample captured video of ESL finger spelling

Image frames Selection

It is obvious that, movie is a sequence of images over time. For instance, the video represented in Figure 2 have 60 frames or images. Some of the frames from the video are depicted in Figure 3. However, all the frames in the video will not use for recognition of the ESL finger spelling. Therefore, the next step is identifying key frames from a set of frames in a given movie.

Figure 3 Sample frame images of ESL finger spelling from a movie

Processing each frame from the movie is not required for the recognition of the ESL finger spellings. In addition to this, processing and analyzing all frames cause delay of processing time and unwanted space usage. Therefore, we have to select smaller number of image frames than the actual number of frames in a given video. For this reason, we use a variable called initial seed variable (IV) for selecting sequence of images and we also applied a distance based selection technique as well. Initial seed variable has a scalar value and it is used to skip frames from the video during processing. It is useful to minimize the processing time. The value for the variable possibly can be one of the numbers {1, 2, 3… 9}. However, the default value is 9; this is the best value to skip number of frames from the movie. The value of the initial variable is limited to the above set. This is because, the required points (center of masses from frames) are four as we discussed latter on the recognition section and in our data we incorporate a one second length. Therefore, if we try to skip fames more than 9, it can produce error because we may select frame out of the total number of frames. The experiment result for the initial variable is depicted on Table I. We experiment 56 videos to represent the seven vowels and we obtain the best recognition result when IV is 6, 7, 8, and 9.

IV	**Vowel 1**	**Vowel 2**	**Vowel 3**	**Vowel 4**	**Vowel 5**	**Vowel 6**	**Vowel 7**	**Total**
1	8	7	8	7	7	4	5	**46**
2	8	7	8	7	7	4	4	**45**
3	8	7	8	7	7	4	5	**46**
4	8	7	8	7	7	4	4	**45**
5	8	7	8	7	7	5	5	**47**
6	8	8	8	7	7	6	4	**48**
7	8	8	8	7	7	5	5	**48**
8	8	8	8	7	7	6	4	**48**
9	8	8	8	7	7	5	5	**48**

Table 1: Experiment results for different IV values

However, the initial seed variable may not detect the motion made by the signer alone. This is because of the variation in speed during signing in a single video. Hence, we used a distance based selection technique as we discussed on motion detection section on Section. Figure 4 shows the general flowchart frame selection algorithm. The algorithm selects key frames to process.

To describe the algorithm in Figure 4, the first image from the set of frames in a given video processed first. This image is possibly used for the recognition of the ESL consonants and it also uses as the first point selected for the recognition of vowels. The consonants can be recognized using the previous study on ESL recognition (Yonas & Raimond, 2010) The next image is in the index 1+ initial seed variable. The rest frames can be selected for process, using initial seed variable or by checking the distance among the previous selected frames. If the distance between previous two adjacent points satisfies the condition for motion occurrence, it uses initial seed variable to select the next frame, otherwise it uses the distance based selection technique.

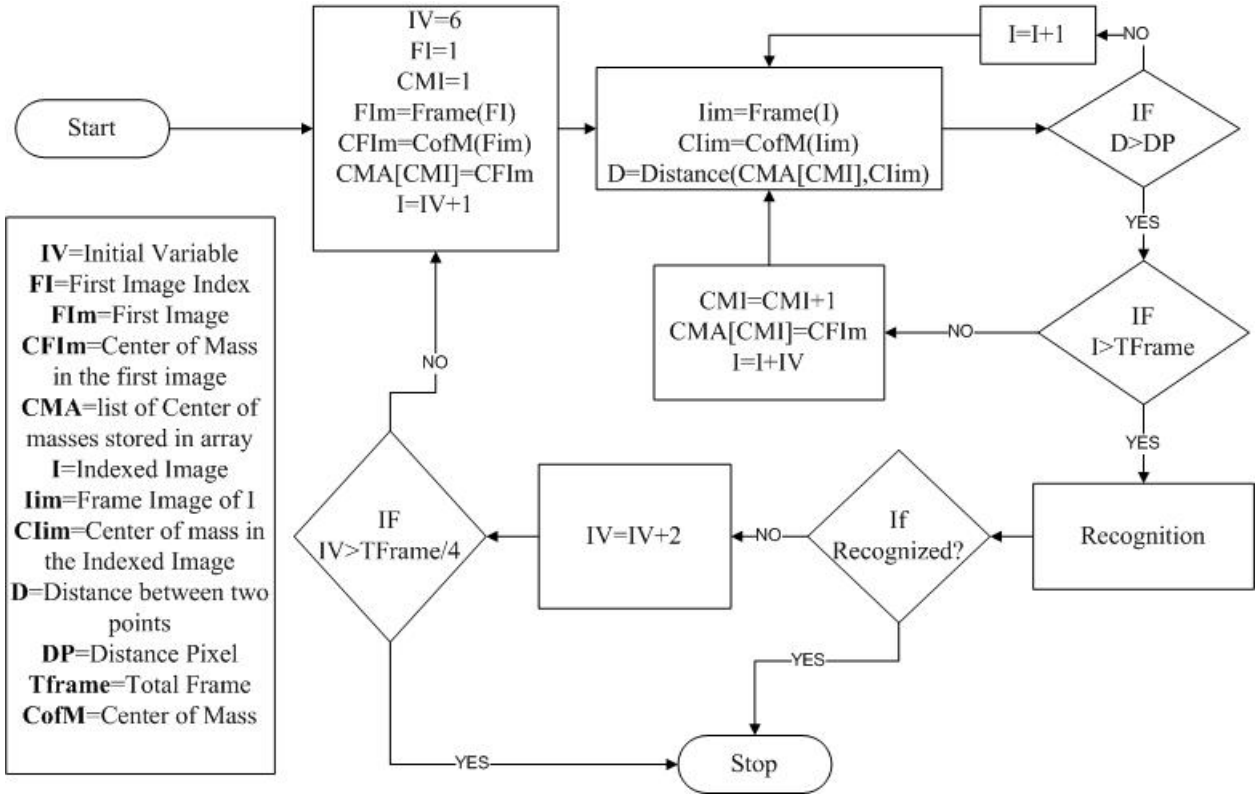
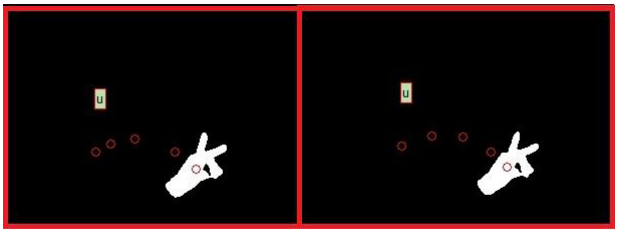


Figure 4 Image selection algorithm from a movie

Distance between selected adjacent points may vary if we only consider the initial seed variable. This happened, because of the speed in a given movie is different. As a result, the system may process frames in the same location or nearby. Distance based selection is essential for having points in different location and approximately uniform distance between points. Three similar distances between adjacent points are used for the recognition of the ESL vowels. This is discussed in the feature extraction section. The frame selection process is used distance based selection technique, if the initial variable will not make a motion. This process can continue by selecting the next frame from the current indexed frame. This process persists until the motion detection assumption is satisfied. The usefulness of initial seed variable and distance based selection is depicted in the Figure 5. In the Figure 5a, the distance between adjacent points doesn't have uniform distance. This is because of the difference in velocity of the signer in a given movie. To make this difference uniform, we use distance based selection technique. In Figure 5b, we observe uniform distance between selected points which represent frames.

Distance based selection uses two adjacent points from two different frames. We find these two points by passing different image processing and analysis techniques like preprocessing, and segmentation of signed hand. As an initial, we use frame one and frame 1+ initial seed variable. Then distance between two points p1(x1, y1), and p2(x2, y2) is computed using Euclidean distance in Equation 1.

$$\square=(\square 2-\square 1)2+(\square 2-\square 1)2 \qquad (1)$$



a) Using only IV b) Using IV and Distance based

Figure 5 Using only Initial seed variable and initial seed variable plus distance based image selection criteria

If the distance between two points satisfies the set condition, it continues to find another point otherwise it calculates distance with the next frame. This process will continue until the end of the number of frames in a given movie. As we can see from Figure 5c, the red color circles are points which are used for the recognition purpose. These circles are points, which satisfy the condition. The blue color circles in Figure 5c show how it processed when the initial variable does not meet the condition. For instance, if we consider the fifth red point counting from the right side, it processed only one frame after the initial variable set. However, in the second point there are a number of frames were processed. This is due to different velocity of the signer in a single movie for a sign.

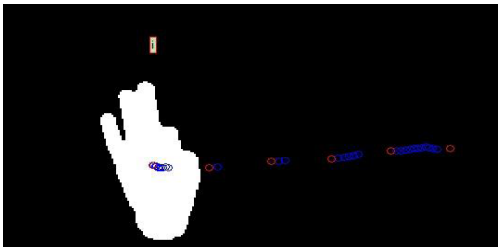


Figure 5c Image selection process from a movie

Finally, the initial variable probably iterates and increases its value. This happened, when all frames in the entire movie were processed and not yet recognized. This process may iterate until the condition IV>Tframe/3 satisfies. This is due to the need of four points for recognition of the ESL as we discussed in on section required points for recognition. If IV>Tframe/3 is true, it produces error. This is because we cannot select a frame, if the index is greater than the total number of frames in a movie. For instance, let us have a signed movie of one second which has 30 frames. If IV is 9, the possible frames used to find the four points can be frame 1, frame 10 frame 19 and frame 28.

However, if IV is greater than or equals to 10, the fourth frame used to find the fourth point will not be in the range. For example, if IV is 10 then possible frames used to find the points are frame 1, frame 11, frame 21, and frame 31. This produced error because frame 31 is not found.

Signed Hand Segmentation

Preprocessing of Images

In this study, the use of preprocessing is to prepare the frames easy for segmenting and isolating sign hand. In the frame selection section, movies are processed and selected image or frame was passed for the next step which is preprocessing of images. The selected images have colored image (combination of Red, Green, and Blue). Hence, every selected image should pass this step and this is essential for the segmentation of the sign hand. On this step, we have the following preprocessing procedures depicted in Figure 6.

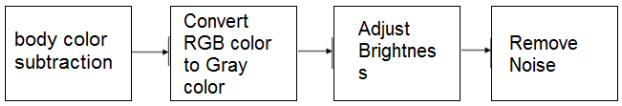


Figure 6. Preprocessing image procedures

To subtract the body color we obtain a single value from the face color. To detect the face color we use Macromedia Dreamweaver application software. Basically, it is possible to use any other application software to obtain the skin color. Algorithm I describes body color subtraction.

Algorithm 1: Human skin color subtraction algorithm Input: Colored or gray image
Output: Skin colored subtracted image

F←Read_image;

R← f(red);

G← f(green)

B← f(blue)

[R G B]←human Skin color; //RGB have scalar values

F←f(r-R,g-G,b-B); // each scalar value is subtracted from corresponding color value image

After each RGB human skin color subtraction is done, the three different images are combined together. But if the images are grayscale the system subtracts the gray value of the human face color.

Segmentation Using Global Thresholding

Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation (Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, 2003). A grayscale image is turned into a binary (black and white) image by first choosing a grey level threshold value (T) from the original image, and then turning every pixel black or white according to whether its grey value is greater than or less than T (Alasdair McAndre, 2004). We already have a gray image from the prior steps, and now this image should be converted to binary image. Suppose that, the gray level image f(x, y) composed of lighted objects on dark background. In order to isolate the lighted objects from the background, we should set some value for T. And using that value it can be grouped in to two using Equation 2.

$$g(x,y)=1 \quad \text{if } f(x,y)\geq T(x,y)0 \quad \text{if } f(x,y)<T(x,y) \qquad (2)$$

The system segments the images into background and foreground using threshold value T. However, choosing threshold value using visual inspection of image histogram or try and error is ineffectual due to the nature of non interactive system. In addition, the chosen threshold value should be used for the entire image. Moreover, the system also needs an automatic choosing of

threshold values. Therefore, the system utilizes an algorithm from Gonzalez and Woods (Rafael C. Gonzalez, et al., 2003) for selecting threshold value iteratively and automatically for global use. Using the algorithm, we convert the images in to binary images, in other words the image converted in to foreground value 1 and background value of 0. However, after segmenting, the foreground image contains objects (noises) less than the size of signed hand. Samples of segmented image with some noise are depicted in Figure 7.

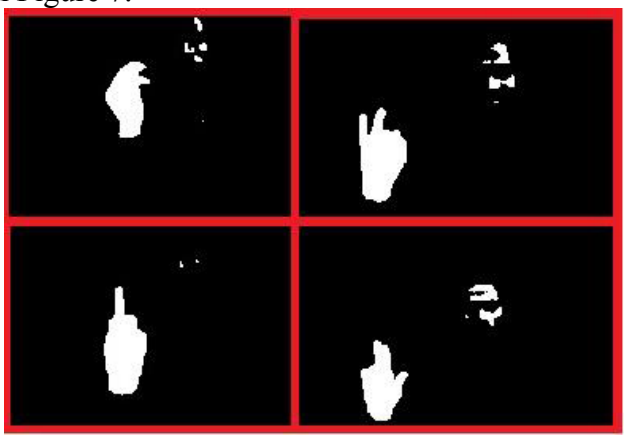


Figure 7. Samples of segmented image after global thresholding

Isolating Signed Hand

Figure 7 shows the result of segmentation using global thresholding. On the other hand, the segmentation process using global thresholding could not isolate the required sign hand. This is occurred since the light effect with the face color makes bright and have high RGB value. Not only the light effect but also small white objects with the signer like cloth buttons and necklace could cause the segmentation process not satisfied yet. However, after segmentation, every created white object with the signed hand is smaller in area size. For this reason, we need a post processing technique for removing these objects; which are smaller in area size than the hand signed.

It is difficult to remove all these unnecessary objects by the common filtering techniques like median filter used in the preprocessing part of this system. To isolate the signed hand from other objects, it is required to group connected objects in the entire image. On the other hand, grouping connected objects is better segmentation technique if we used 4-connected neighborhood. Grouping 4-connected neighborhood is useful to isolate objects connected by the corner side of the adjacent pixel. After grouping each different object from the entire image, the largest object in area is selected for the next process. This means, the connected block with maximum number of pixels are selected as a sign hand. The algorithm used to segment this object is here in Algorithm 3.

Algorithm 2: Algorithm for isolating hand sign

Input: Segmented image using global thresholding, it has other objects with the sign hand

Output: Isolated sign hand image

1. Get image from segmentation process
2. Group connected objects using 4-connected neighborhood
3. Calculate pixel size of the regions
4. Select the largest area
5. Put the selected area on its location with its previous frame size

After selecting the largest area which is the sign hand, all unnecessary objects in the entire image is removed. Only the signed hand is presented with its previous frame size. Figure 8a, and Figure 8b shows before and after post processing respectively. The use of putting the image to the previous frame size on its location is useful for the recognition of vowels of ESL. However, the selected image used for the consonant recognition can be resized according to the previous study on ESL recognition (Yonas & Raimond, 2010).

a) b)

Figure 8 Isolated signed hand

Feature Extraction

From the segmented signed hand, the vital information is a single dot which is the center mass of the object.

Computing Center of mass

To detect the motion of the frames, it is necessary to take a reference point of each segmented image. Therefore, we use center of mass of the segmented sign hand. The center of mass is the mean location of all the masses in a given environment. In other words, the center of mass is the point at which you can balance all the objects. The law of center of mass is defined by Equation 3:

□=□=1□□□□□□□=1□□□ (3)

Where Mi is for objects or masses and Ri is distance

In our case, the objects that we are assuming as individual masses are the number of pixels from an entire image. In addition, every object in the given image has one of the two values, which is the foreground weight of 1 and background weight 0. The objects also represented in a two dimension plane, x and y. For two dimensional image f, the pixels values are represented by f(x, y). Therefore, the mean location for Rx and Ry are:

$$R_x = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} i*f(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)} \quad (4)$$

$$R_y = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} j*f(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j)} \quad (5)$$

Therefore the center mass is:

□=□(□□,□□) (6)

Based on Equation 6, the algorithm used for finding center of mass is presented in Algorithm 3.

Algorithm 3: Algorithm for extract features

Input: Isolated signed hand, f (x, y)

Output: Point that represents the center mass of the sign hand

1. Read image f
2. Rx←0, Ry←0, n←0
3. Find w←Width of f
4. Find h←Height of f
5. For i=1 to w
6. For j=1 to h

7. If f(i, j) ←foreground image // if 1
8. Rx=←Rx+i
9. Ry←Ry+j
10. n←n+1
11. End if
12. End for
13. End for
14. Rx← Rx/n
15. Ry←Ry/n
16. R←f(Rx,Ry)

Some of the center of masses from the set of frames for the ESL manual alphabets is depicted in Figure 9.

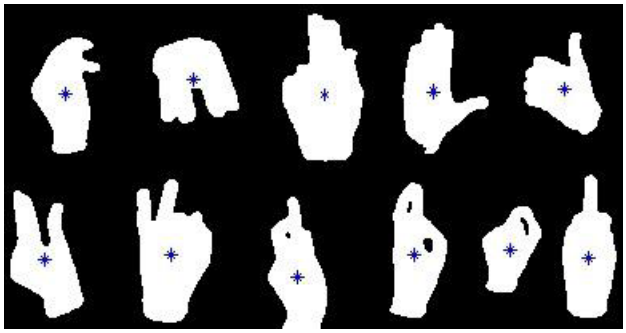


Figure 9 Sample of Center of masses of the ESL manual alphabets

Recognition

Motion Detection

The ESL finger spelling integrates hand form with movements. Hence, motion detection is essential before recognizing the ESL finger spellings. According to [7, 6], ESL manual alphabets have seven movements corresponding to the seven Amharic vowel orders. The first vowel order (Geez) is a motionless, others have six special movements. Therefore, to understand these movements it is practical to detect the motion state.

In order to identify these movements, first it is decisive to decide the length of the distance LD, which is an assumption for no movement. The movement made by unconsciously hand shaking of the signer, error occurred in image processing, unconsciously body movement of the signer and camera vibration are assumed to be the motion is in a motionless state.

LD= unconsciously hand shaking + error in image processing + camera vibration + unconsciously body movement

To choose the length of LD, we used the maximum distance between the initial frame and others in a given movie of vowel-1 (Geez). Even if the first vowel is represented by stationary motion, there are motions made by LD. Therefore, we have to assume the motion made less than or equals to LD is in motionless. The algorithm used to find LD is presented in Algorithm 4. During motion detection, we always calculate the center of mass from the first frame and it is used as a first point for recognition. Hence, the first frame is the reference point to detect the motion. Therefore, in order to assume the motion made by LD, the starting point is frame one. However, during finding LD we cannot assume that the maximum distance is between frame one and last frame. This is because of the motion is made by a kind of vibration. This is the reason why we need algorithm to find the LD.

Algorithm 4: Finding distance LD

Input: Video Clip for vowel-1 (Geez)

Output: The Maximum distance between frame 1 and others

1. Read Video V_Geez

2. Find the total number of frames of V_Geez, Tframe

3. LD←0

4. For I=2 to Tframe

5. Find distance, D =Distance(Frame 1, Frame I)

6. If D>LD Then

7. LD=D

8. End if

9. I←I+1

10. End For

11. Stop

Using Algorithm 4, we test all of the ESL vowel-1. As a result, we obtain 34 different lengths of LD representing motion made by the 34 ESL vowel-1 finger spellings. From the experiment result, we select the maximum one for the actual LD. From our data, we obtained the minimum value of LD is 2.7869(pixels), the maximum value of LD is 16.8525 and average value of LD is 8.3247.

Required Points for Recognition

To recognize the seven movements, we use points that represent frames. In our case, the seven vowels of Ethiopian Sign language are characterized by at least three connected lines or four points. This is because, if the number of points is less than four points, it cannot differentiate all movements and if it is greater than four points it has an impact on decreasing the processing time. These points are represented in terms of x and y directions. In this case, P1 is represented by (x1, y1) which is the center of mass of frame 1 from the given movie, P2 is (x2, y2), P3 is (x3, y3), and P4 is (x4, y4).

Now it is essential to have an ideal machine, that takes properties of these points and produce the corresponding vowel number representation. Therefore, automata are our choice in this case. To understand the vowel or movements, we use a non deterministic finite state automaton.

A non-deterministic finite state of automaton is defined by: $M= (Q, \Sigma , \partial, q0, F)$

Q- Finite set of internal states

Σ-Finite set of input alphabets

∂- $Q \times (\Sigma \cup \{ \lambda \}) \rightarrow 2Q$, where λ is an empty string

$q0$- is the initial state $q0 \in Q$

F- is a set of final states $F \subset Q$

The finite set of alphabets used in the recognition of ESL is derived from LDx, LDy and the four points. To find the required alphabets, it is necessary to find the difference between adjacent points. Accordingly, X21=x2-x1, X32=x3-x2, X43=x4-x3, Y21=y2-y1, Y32=y3-y2, and Y43=y4-y3. From this, movements we considered when X21> LDx, X32> LDx, X43> LDx, Y21> LDy, Y32> LDy, Y43> LDy, X21<- LDx, X32<- LDx, X43<- LDx, Y21<- LDy, Y32<- LDy, and Y43<- LDy. Along with, movements will not consider when abs(X21) LDx, abs(X32) LDx, abs(X43) LDx, abs(Y21) LDy, abs(Y32) LDy, and abs(Y43) LDy. Therefore, we have 18 input alphabets for the recognition of the vowels. The minus (-) signs are used for the left and down direction. Table II shows the 18 inputs alphabets are grouped into six directions from the screen point of view.

Left direction	Right direction	Up direction	Down Direction	In stationary X-axes	In stationary Y-axes
X21<-LDx, X32<-LDx, X43<- LDx	X21> LDx, X32> LDx , X43> LDx	Y21<- LDy, Y32<- LDy, Y43<- LDy	Y21>LDy, Y32>LDy, Y43> LDy	abs(X21)≤ LDx, abs(X32)≤ LDx, abs(X43)≤ LDx	abs(Y21)≤ LDy, abs(Y32)≤ LDy, abs(Y43)≤ LDy

Table 2 Possible input alphabets grouped in to six directions

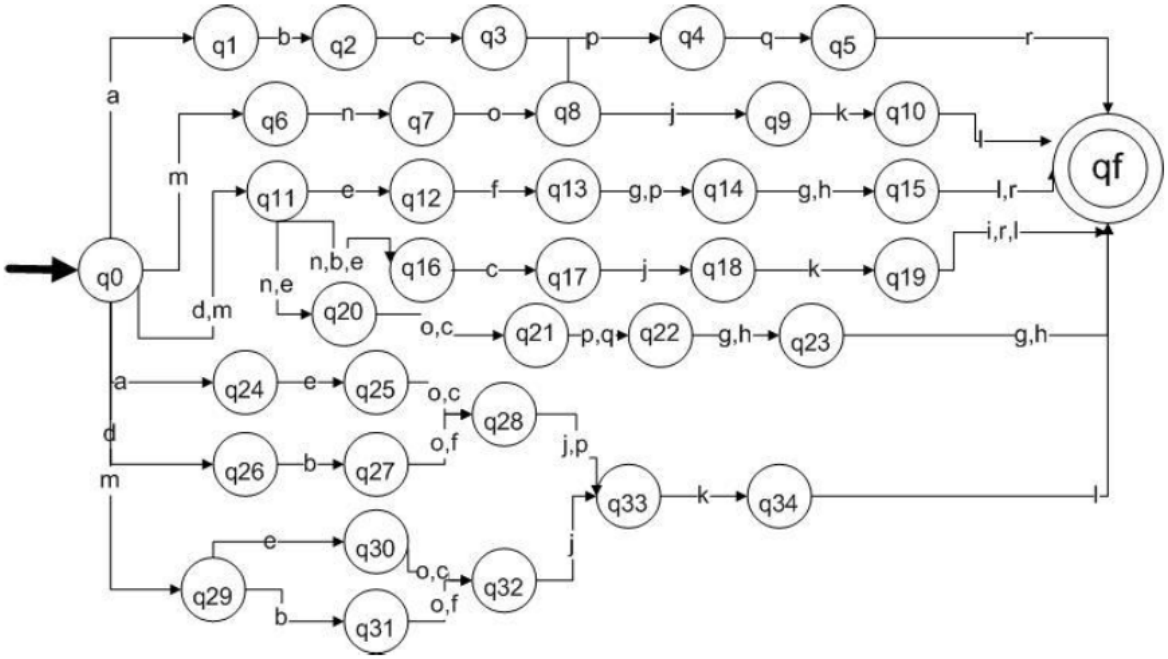
To form a general finite state of automata, it is important to discus strings that can generate vowels using the 18 input alphabets. For simplification reason, we symbolize the possible input conditions as shown in Table 3.

Letter	Condition	Letter	Condition	Letter	Condition	Letter	Condition
A	X21<- LDx	f	X43> LDx	k	Y32> LDy	p	abs(Y21)≤ LDy
B	X32<- LDx	g	Y21<- LDy	l	Y43> LDy	q	abs(Y32)≤ LDy
C	X43<- LDx	h	Y32<- LDy	m	abs(X21≤ LDx	r	abs(Y43)≤ LDy
D	X21> LDx	i	Y43<- LDy	n	abs(X32)≤ LDx		
E	X32> LDx	j	Y21> LDy	o	abs(X43)≤ LDx		

Table 3 Symbolize input alphabets by small letters

As a final point, the automata used to recognize the ESL vowels defined by:

M=({q0,q1,q2,q3,q4,q5,q6,q7,q8,q9q10,q11,q12,q13,q14q15,q16,q17,q18,q19,q20,q21, q22,q23,q24q25,q26,q27,q28,q29 q30,q31, q32,qf}, {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r}, ∂, q0, {qf}), ∂ defined by the following transition graph:



EXPERIMENT AND RESULT

In this research, we obtain the signs using a single digital camera with the help of white glove. The signer wears white glove for a reason of easy segmentation. The camera is taken by a person while the signers stand with black background. The camera is set in front of the signer. The data used for this experiment is collected by four signers. The data is collected in terms of video mode. All signers spelled together all of the 238 ESL finger spellings. A single movie is represented to a single sign. Consequently, the 238 ESL manual alphabets are incorporated in our data. Furthermore, the data is clustered using the seven vowels of the ESL finger spelling and evaluated the performance of the vowels. Each clustered vowels have 34 signed videos.

The seven movements of ESL vowels are assigned to a value of one, two up to seven numbers. The validation of the recognition of the system works by crosschecking the vowel with its file name of the movie. If the result of the recognition of the sign is equal with the file name of the movie, then it counts as recognized otherwise it counts us unrecognized vowel.

The experiment is done to evaluate the recognition performance of the new method. We can see this using the overall recognition performance of the system, recognition performance of each vowel group and recognition performance per signer. The overall recognition performance of the system is calculated using the Equation 7.

$$\text{System Performance} = (\text{Total Number of Recognition of Vowel})/238 \times 100 \quad (7)$$

Accordingly, the overall system recognition performance is 90.75 %. This means 216 finger spellings are recognized from total of 238 finger spellings. The performance of the clustered vowels is calculated using Equation 8. The recognition performance results of each clustered vowel are listed in Table 4.

$$\text{Vowel clusterd Performance} = (\text{Total Recognition of Vowel Clusterd})/34 \times 100 \quad (8)$$

As we can see from Table IV vowel 1, vowel 2, vowel 3, vowel 4, and vowel 5, have excellent recognition accuracy. Vowel 6 and Vowel 7has also satisfactory result but errors occur due to error in signing inputs. The move for vowel -6 is zigzag, and then signers can make this simple vibration of hand. Consequently, the assumption for movement of sign could not detect the motion in the x-axis. In case of Vowel-7, due to the input of some signs are rotated on its fixed point and created similarity with vowel-1-. However, if the data could be collected in a more accurate way and if the signers strictly follow the rule how to sign, better result can be obtained.

Vowels	Total Number of Vowel	Recognized Vowels	Recognized Percentage (%)
Vowel-1-	34	33	97.0588
Vowel-2-	34	33	97.0588
Vowel-3-	34	32	94.1176
Vowel-4-	34	31	91.1765
Vowel-5-	34	32	94.1176
Vowel-6-	34	28	82.3529
Vowel-7-	34	27	67.4116

Table 4 Experimentation Result of vowels

Conclusion

In this study we design and develop a method used to recognize vowels of Ethiopian sign language from a video. In this work we applied image preprocessing algorithms, image segmentation, and post processing before trying to understand the signs. In addition, we used image selection algorithm to increase the efficiency of the system. Preprocessing and post processing are used for accurate recognition. We also used image segmentation to identify the sign hand from the entire image.

To recognize the vowels of ESL, we extract features from the segmented sign hand. The extracted feature from the entire image is the center mass of the signed hand. The extracted information was used to detect the motion and to recognize the vowels. Group of center masses are finally used for the recognition of vowels. To recognize these vowels, we employed finite state automata.

For the successful completion of the study, we collect all of the 238 ESL figure spellings by four signers in terms of video mode. The data is experimented using our system. As a result, the overall system achieved 90.75% recognition performance. Therefore, it is possible to conduct projects and researches for more work on the language by using this study as a base. Moreover, the thesis has its own role on the study of motion detection and image processing applications.

References

Alvi, A. K., Bin Azhar, M. Y., Usman, M., Mumtaz, S., Rafiq, S., Razi, U. R., & Ahmed, I. (2005). *Pakistan Sign Language Recognition Using Statistical Template Matching.* Proceeding of the World Academy of Science, Engineering and Technology, Las Cruces, USA.

Alasdair McAndre. (2004). *An Introduction to Digital Image Processing with Matlab*", Notes for Digital Image Processing. Retrieved from http://visl.technion.ac.il/ labs/anat/An Introduction To Digital Image Processing With Matlab.pdf

Duarte, K. (2010). The Mechanics of Fingerspelling: Analyzing Ethiopian Sign Language. Sign Language Studies, 11(1), p. 5-21.

Ethiopian National Association of the Deaf. (2008). Ethiopian sign language dictionary, Addis Ababa, Ethiopia.

National Interpreter Resource Links. (2011). *Ethiopian Manual Hand Alphabets.* Retrieved from httpwww.terpslink.net\WSL\EtSL.html

Rafael C., Gonzalez, Woods, R. E., & Eddins, S. L. (2003). *Digital Image Processing using Matlab*. Prentice Hall.

Ricco, S. and Tomasi, C. (2009). *Fingerspelling Recognition through Classification of Letter-to-Letter Transitions.* In Proceedings of ACCV (3), p.214-225.

Tadesse, M. (2010). *Automatic translations of Amharic text to Ethiopian sign language.* A thesis submitted to the school of graduate studies of Addis Ababa University.

Vassilia N. & Konstantinos G. (2002). *Hidden Markov Models for Greek Sign Language Recognition.* Proceedings of 2nd WSEAS International Conference on Speech Signal and Image Processing ICOSSIP, Skiathos, Greece.

Yonas A., & Raimond K. (2010). *Ethiopian sign language recognition using Artificial Neural Network.* In Proceedings of the international conference on Intelligent Systems Design and Applications (ISDA) IEEE, p.995-1000.