SIGN GESTURE SPOTTING IN AMERICAN SIGN LANGUAGE
USING DYNAMIC SPACE TIME WARPING

by

SRUJANA GATTUPALLI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MS in Computer Science

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2013

Acknowledgements

I would like to convey my warmest gratitude to my supervising professor Dr. Vassilis Athitsos for giving me the opportunity to conduct research on this project and for his constant guidance, support and encouragement from the start until the end of my study. I wish to thank my committee members Dr. Farhad Kamangar and Dr. Gian Luca Mariottini for their interest in my research and for taking time to serve in my dissertation committee.

I would like to thank Ms. Carolyn Stem for teaching me American Sign Language that helped me to get a better insight in my research work.

I would also like to extend my appreciation to CSE department for providing all the facilities and infrastructure necessary to carry out my Master's studies. Special thanks to all my teachers in India and in United states who have taught me and helped me achieve the knowledge and education that I have today.

I would like to thank my beloved parents, who have taught me the value of hard work by their own example. I would like to thank all my friends for their affection and care that brings color to my life.

Finally, I would like to thank all whose direct and indirect support has helped me in completing my thesis in time.

April 11, 2013

Abstract

SIGN GESTURE SPOTTING IN AMERICAN SIGN LANGUAGE USING DYNAMIC SPACE TIME WARPING

Srujana Gattupalli, MSc

The University of Texas at Arlington, 2013

Supervising Professor: Vassilis Athitsos

American Sign Language (ASL) is the primary sign language used by approximately 500,000 deaf and hearing-impaired people in the United States and Canada. ASL is a visually perceived, gesture-based language that employs signs made by moving the hands combined with facial expressions and postures of the body. There are several software tools available online to learn signs for a given word but there is no software that gives the meaning of any sign video. If we need to search or look up documents containing any word we can just type it in search engines like Google, Bing etc. but if we need to search for videos containing a given sign we cannot do that simple by typing a word. One solution to this problem can be adding English tags to each of these ASL videos and do a keyword based search on it. This method can be inefficient as each sign video needs to be tagged manually with approximate English translations for each of the ASL gesture. The objective is to develop a system that lets users efficiently search through videos of ASL database for a particular sign. Given an ASL story dataset the user can use the system to know the temporal information (start and end frame) about the occurrence of the sign in the dataset. Recognizing gestures when the start and end frame of the gesture sign in the video is unknown is called Gesture spotting. The existing system evaluates the similarity between the dataset video and the sign video database using Dynamic Time Warping (DTW). DTW measures similarity between two sequences which may vary in time or speed. In this paper we have used a previously defined similarity measure called Dynamic Space Time Warping Algorithm (DSTW). DSTW was defined as an extension of DTW in order to deal with a more than one hand candidate by frame. This provides a method to find a better optimal match by relaxing the assumption of correct hand detection. This paper contributes by establishing a baseline method for measuring state of the art performance on this problem. We have performed extensive evaluations of DSTW on a real world dataset. We have achieved this by implementing a DSTW gesture spotting algorithm and evaluating it on a dataset built on ASL stories. We have used transition costs in DSTW to improve accuracy. We have also performed evaluation of accuracy based on motion scores for various signs. We have used single handed queries from the ASL dictionary and used a symmetric rule for evaluation of the classification accuracy of our system for each of these queries.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

American Sign Language (ASL) is used by 500,000 to two million people in the U.S. [1]. Many resources that are taken for granted by users of spoken languages are not available to users of ASL, given its visual nature and its lack of a standard written form. One such resource is the ability to look up the meaning of an unknown sign.

[3] explains on how difficult it is to look up a meaning of an unknown sign. It is not a straightforward task as looking up a word from a written language in a dictionary. A system that helps users look up unknown signs would be useful to the millions of users and learners of sign languages around the world. [3] presents a method to help users look up the meaning of an unknown sign from ASL. The user perform the sign in front of a webcam and submits a video of the unknown sign as a query. Then, the system compares the input sign with videos of signs stored in the system database, and presents the most similar signs (and potentially also their English translations) to the user. The user can then view the results and decide which (if any) of those results is correct.

[4] works with [3] on including video examples from a vocabulary that is similar in scale and scope to the set of lexical entries in existing ASL-to-English dictionaries. [4] uses at least one video example per sign from a native signer, for almost all of the 3,000 signs contained in the Gallaudet dictionary. Signs are differentiated from one another by hand shape, orientation, location in the signing space relative to the body, and movement.

This paper improves on [4], [3] by providing a protocol for a user to look up a sign from a dataset video that comprises of multiple ASL signs. Given an ASL story dataset the user can use the system to know the temporal information (start and end frame) about the occurrence of the sign in the dataset.

The existing ASL system [4],[3], [1] uses location of hands and face for sign recognition. These locations have been manually annotated by [1].

ASL Lexicon Video Dataset [1],is a large and expanding public dataset containing video sequences of thousands of distinct ASL signs, as well as annotations of those sequences, including start-end frames and class label(English translation) of every sign. This dataset is being created as part of a project to develop a computer vision system that allows users to look up the meaning of an ASL sign.

In natural settings, hand locations can be ambiguous in the presence of clutter, background motion and presence of other people or skin-colored objects.[5]. Furthermore, gestures typically appear within a continuous stream of motion, and automatically detecting where a gesture starts and ends is a challenging problem. To recognize manual gestures in video, an end-to-end computer vision system must perform both spatial and temporal gesture segmentation. Spatial gesture segmentation is the problem of determining where the gesture occurs, i.e., where the gesturing hand(s) are located in each video frame. Temporal gesture segmentation is the problem of determining when the gesture starts and ends. Recognizing gestures when the start and end frame of the gesture in the video are unknown is called Gesture spotting.

This paper works on the ongoing project [5] to form a unified framework that accomplishes all three tasks of spatial segmentation, temporal segmentation, and gesture recognition, by integrating information from lower-level vision modules and higher-level models of individual gesture classes and the relations between such classes. Instead of assuming unambiguous and correct hand detection at each frame, the proposed algorithm makes the much milder assumption that a set of several candidate hand locations have been identified at each frame. Temporal segmentation is completely handled by the algorithm, requiring no additional input from lower level modules

[3],[4] define feature vectors based on hand motion and hand appearance. Similarity between signs is measured by combining dynamic time warping scores, which are based on hand motion, with Euclidean distances between hand appearances of the dominant hand for the query sign. The similarity between the query video and the sign video database is evaluated using Dynamic Time Warping (DTW). DTW measures similarity between two sequences which may vary in time or speed. DTW algorithm finds an optimal match between two time series with certain restrictions. These time sequences are the trajectories based on hand locations for every frame of the video for the query and the sign database.

The paper attempts to address the problem of Gesture spotting using Dynamic Space Time Warping Algorithm (DSTW). DSTW was defined in [7] as an extension of DTW in order to deal with a more than one hand candidate by frame. This provides a method to find a better optimal match by relaxing the assumption of correct hand detection.

[7] proposed an approach which is a principled method for gesture recognition in domains where existing algorithms cannot reliably localize the gesturing hand. Instead of assuming perfect hand detection, we make the milder assumption that a list of candidate hand locations is available for each frame of the input sequence. At the core of our framework is a dynamic space-time warping (DSTW) algorithm that aligns a pair of query and model gestures in time, while at the same time it identifies the best hand location out of the multiple hypotheses available at each query frame.

The proposed framework is demonstrated in two gesture recognition systems. The first is a real-time system for continuous digit recognition, where gestures are performed by users wearing short sleeves, and (at least in one of the test sets) one to three people are continuously moving in the background. The second demonstration system enables users to find gestures of interest within a large video sequence or a database of such sequences. This tool is used to identify occurrences of American Sign Language (ASL) signs in video of sign language narratives that have been signed naturally by native ASL signers.

Chapter 2

Related Work

This paper works with [5] to devise a method that differs from existing work in gesture recognition and requires neither spatial nor temporal segmentation to be performed as preprocessing. In many dynamic gesture recognition systems lower-level modules perform spatial and temporal segmentation and extract shape and motion features. Those features are passed into the recognition module, which classifies the gesture. In such bottom-up methods, recognition will fail when the results of spatial or temporal segmentation are incorrect.

The existing American Sign Language project [4], [1], [3] has annotated the video signs and tried to improve the similarity measure by using additional spatio-tempotral information available in the video frames. The underlying similarity measure used for sign recognition is Dynamic Time Warping(DTW).

2.1 DTW Algorithm

The existing American Sign Language project [4], [1], [3] has annotated the video signs and tried to improve the similarity measure by using additional spatio-tempotral information available in the video frames. The underlying similarity measure used for sign recognition is Dynamic Time Warping(DTW) (Kruskal and Liberman, 1983).

The DTW algorithm temporally aligns two sequences, a query sequence and a model sequence, and computes a matching score, which is used for classifying the query sequence. Each sign video X is represented as a time series(X_1,...,$X_{|X|}$), where $|X|$ is the number of frames in the video. Each X_t, corresponding to frame t of the video, is a 2D vector storing the (x, y) position of the centroid of the dominant hand, for one-handed signs, or a 4D vector storing the centroids of both hands, for two-handed signs. In particular, let Q be a test video and X be a training video. A warping path W=($(w_{1,1},w_{1,2}),\ldots,(w_{|W|,1}, w_{|W|,2})$) defines an alignment between Q and X. The i-th element of W is a pair ($w_{i,1}$, $w_{i,2}$) that specifies a correspondence between frame $Qw_{i,1}$ of Q and frame $X_{wi,2}$ of X. Warping path W must satisfy the following constraints:

- Boundary conditions: $w_{1,1}= w_{1,2}= 1$, $w_{|W|,1}=|Q|$ and $w_{|W|,2}= |X|$.
- Monotonicity: $w_{i+1,1}-w_{i,1}\geq 0$, $w_{i+1,2}-w_{i,2}\geq 0$.
- Continuity: $w_{i+1,1}-w_{i,1}\leq 1$, $w_{i+1,2}-w_{i,2}\leq 1$.

For one-handed signs, the cost C(W) of the warping path W is the sum of the Euclidean distances between dominant hand centroids of corresponding frames Qwi,1 and Xwi,2. For two-handed signs, we include in the cost C(W) the sum of the Euclidean distances between non-dominant hands in corresponding frames. The DTW distance between Q and X is the cost of the lowest-cost warping path between Q and X, and is computed using dynamic programming (Kruskal and Liberman, 1983), with time complexity O(|Q||X|).

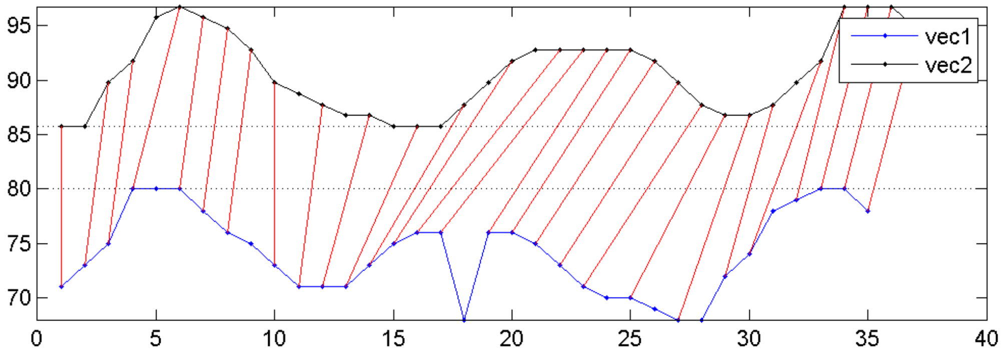


Figure 2-1 Meaningful Alignment

Red Lines Connect Corresponding Points of the Two Vectors

2.2 Improved similarity measure

In DTW, it is assumed that a feature vector can be reliably extracted from each query frame. However, this assumption is often hard to satisfy in vision-based systems, where the gesturing hand cannot be located with absolute confidence. Problems can be caused by a variety of factors, such as changing illumination, low quality video and motion blur, low resolution, temporary occlusion, and background clutter. [7] proposes a state-of-the-art video-based similarity measure called Dynamic Space-Time Warping (DSTW) for the purposes of sign recognition.

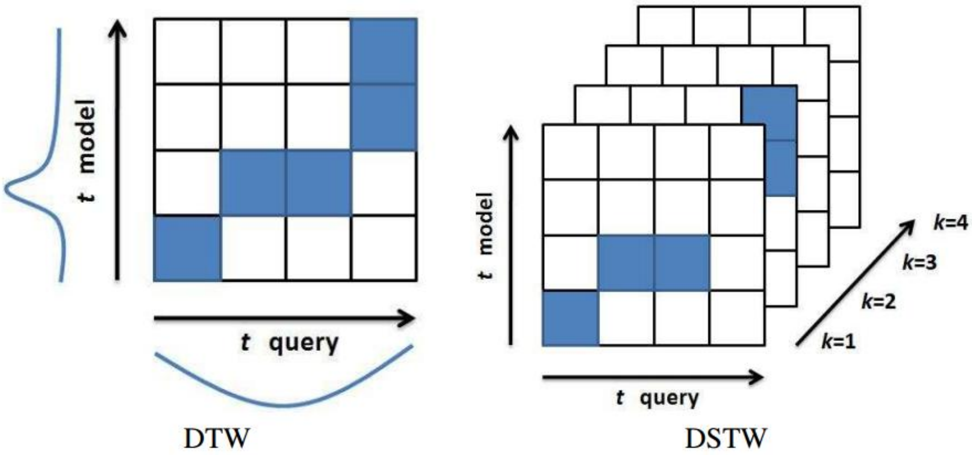


Figure 2-2 Example of Dynamic programming using DTW (left) and DSTW (right)

We adapt DSTW so as to tolerate differences in translation and scale. DSTW is an extension of the popular Dynamic Time Warping (DTW) similarity measure for time series. Given a dataset video, instead of requiring the hand locations to be specified (as DTW does), DSTW makes the much milder assumption that a hand detection module has produced a relatively short list of candidate hand locations. The ability to handle multiple candidate hand locations allows DSTW to be seamlessly integrated with existing imperfect hand detectors. As a result, DSTW-based systems can be readily deployed for gesture recognition in challenging real-world environments.[6]

'Gesture spotting' is a method for recognizing gestures in continuous streams in the absence of known temporal segmentation. This paper works with [5] in developing a gesture spotting method that does not require the hand to be unambiguously localized at each frame of the video sequence. We do not assume known temporal segmentation nor accurate hand detection. We include a much wider array of experiments, including experiments with challenging backgrounds (including two or three moving people acting as distracters), and experiments on a new computer vision application, i.e., ASL sign retrieval in a video database of native ASL signing.

Chapter 3

Contribution

The existing American Sign Language project in UTA has annotated the video signs and tried to improve the similarity measure by using additional spatial(where the gesture occurs, i.e., where the gesturing hand(s) are located in each video frame) and temporal(when the gesture starts and ends) information available in the video frames. Our method requires neither spatial nor temporal segmentation to be performed as preprocessing.

In this paper we have used a previously defined similarity measure called Dynamic Space Time Warping Algorithm (DSTW). DSTW finds a better optimal match by relaxing the assumption of correct hand detection.

Our contributions are as follows:

- We have performed extensive evaluations of DSTW on a real world dataset. We have used transition costs in DSTW, to improve accuracy.
- We have established a baseline method for measuring state of the art performance on this problem.
- We have achieved this by implementing a DSTW gesture spotting algorithm and evaluating it on a dataset built on ASL stories.
- Analysis of motion in ASL sign gestures for improving performance accuracy: What works well and what does not?

Chapter 4

The Dataset

We evaluate the proposed gesture recognition framework in two settings: a gesture recognition setting, where users perform gestures corresponding to the 10 digits and an ASL sign retrieval setting.

4.1 Digits Dataset

Users are asked to make gestures chosen from the 10 Palm Graffiti Digits, as shown in Figs. 3.1.The training set used for learning the digit models consisted of 30 Video sequences from uncompressed training dataset, 3 sequences from each of 10 users. . Users wore color gloves (green) in the training sequences. Each video has 10 gestures from 0 to 9 at annotated positions with green hands. . The 30 training sequences were used in the offline learning stage, to learn models, pruning classifiers, and subgesture relations.

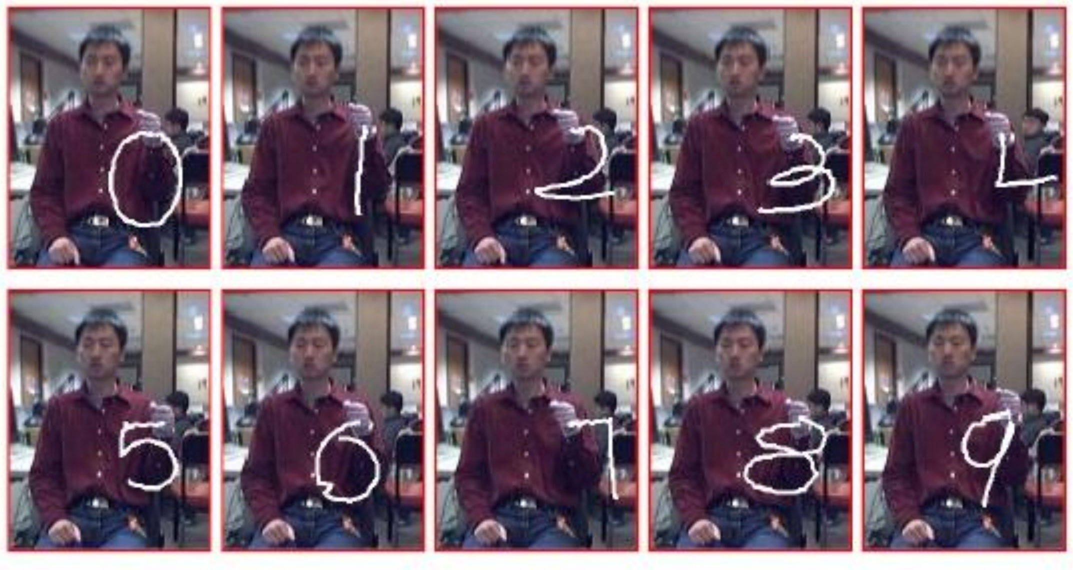


Figure 4-1 Example Palm's Graffiti Digit gestures

We used two test sets for evaluating performance on digit recognition: an "easy" set and a "hard" set. In each test sequence, the user signed once each of the 10 digits, and wore short sleeves (and, naturally, no color gloves). For both test sets, experiments were performed in a user-independent fashion, where the system recognizes test gestures of a particular user using digit models which were learned from training examples collected from other users.

4.2 ASL Narratives Database

The ASL Narratives Dataset consists of videos of sign language narratives that have been signed naturally by native ASL signers. The videos are classified according to the stories contained within them. The dataset characteristics are shown in Table 3.1.

Table 4-1 ASL Narratives Dataset

Video Class (Story)	Video Length in Frames	Video Size (GB)	Video length (Minutes)
Accident	16,608	14.4	4.6
Biker_Buddy	4,689	4.11	1.3
Boston	26,741	23.00	7.4
Dorm_Prank	11,592	11.11	3.2
Football	12,901	11.12	3.6
LAPD_Story	18,009	15.54	5
Muhammed_Ali	9,275	8.09	2.6
Roadtrip_1	17,163	14.8	4.8
Roadtrip_2	11,412	9.84	3.2
Scary_Story	16,064	14.01	4.5
Siblings	10,420	9.06	2.9
Whitewater_Rafting	11,958	10.42	3.3
Total	**166,832**	**145.5**	**46.4**

This dataset is annotated with the signs occurring in it. These annotations are used to evaluate results of experiments. Sometimes one English word can have multiple ASL signs corresponding to it and sometimes an ASL sign can be interpreted as different English words. We have manually compared each sign of ASL Narratives dataset with its corresponding sign in the ASL Lexicon Video Dataset[1] to ensure that the ASL signs and their English meanings are the same. The ASL Lexicon Video Dataset is to eventually contains examples of almost all of the 3,000 signs contained in the Gallaudet dictionary.

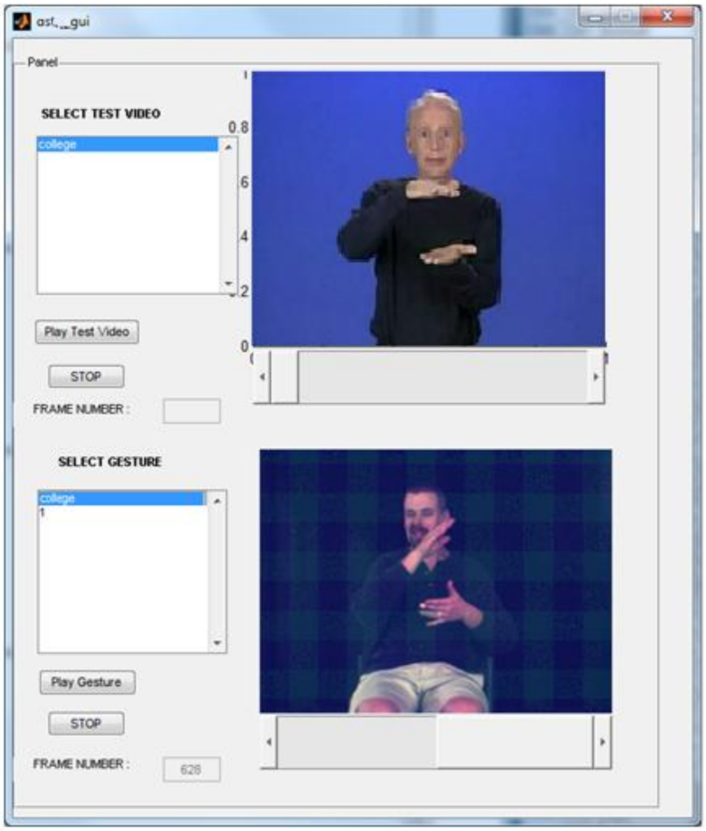


Figure 4-2 GUI for observing gestures from ASLLVD[1] (top video) and ASL Narrative stories(bottom video)

The video playing in first axes is the ASL sign for the word “college” in ASL dictionary. The video in second axes is the ASL sign for “college” which appears in the ASL Narrative video 1 of LAPD_story dataset. We observed that it is similar to the sign in the dictionary. In Fig. 3.2 the list box on the left of the lower axes shows multiple occurrences of same sign. Frame number is shown at the left bottom corner. Similarly we observed gestures for other signs that we used for our experiments.

Chapter 5

Problem Definition and Overview of Method

The user performs a gesture or inputs the rough English translation for an ASL gesture sign and selects the video from the ASL narrative dataset that he want to find the gesture from. Our method assumes that we neither know the location of the gesturing hands in each frame, nor that we know the start and end frame for every gesture.

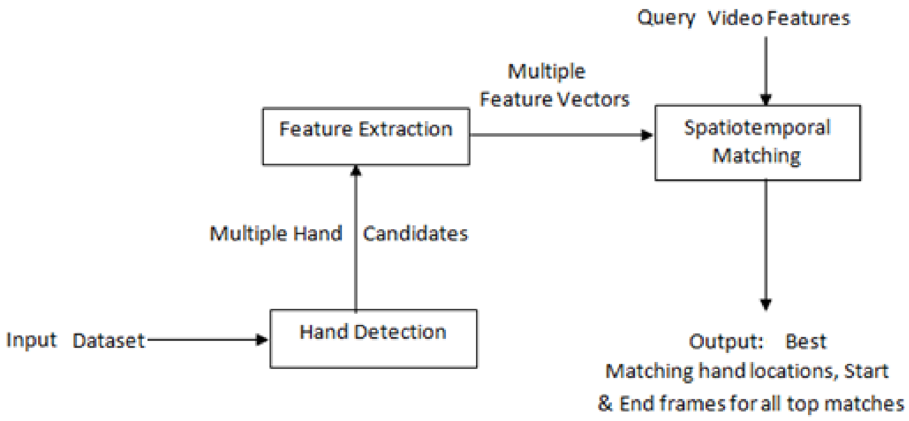


Fig. 5-1 Gesture Recognition system flowchart

The gesture recognition algorithm consists of components for hand detection and feature extraction, spatiotemporal matching, obtaining the optimal warping path. The overall algorithm is depicted in Fig. 5.1.

Feature extraction element is used in our experiment. Multiple candidate hand regions are detected using size, skin color and motion. Each candidate hand region is a rectangular image sub window, and the feature vector extracted from that sub window is a 4D vector containing the centroid location and optical flow of the sub window. The algorithm basically focuses on movement of the hand and the location of the hand in the frame.

Once the signs have been ranked, the system presents to the user an ordered list of the best matching signs. The user then views the results, starting from the highest-ranked sign, until encountering the video displaying the actual sign of interest. For example, if the correct match was ranked as 10th best by the system, the user needs to view the top ten results in order to observe the correct match.

Our algorithm has been used for one handed signs for the ASL dataset. In certain videos of the dataset the gesture occurs multiple times and in some there is only single occurrence for a gesture. The algorithm shows rank of all the gestures matches that it finds in the video dataset. We have evaluated the results using precision and recall. The system decides whether a gesture occurs or not using the symmetric distance algorithm which is explained in Chapter 8. The result from the gesture recognition algorithm is input to the symmetric distance algorithm which already knows the annotations of our dataset and decides whether a gesture is recognized correctly.

Chapter 6

Hand Detection and Feature Extraction

Hand detection is used to compute centroids for the Dataset and the Query sign videos to obtain the corresponding time series. These time series are then used by the DSTW algorithm to get the similarity measure between the Dataset and the Query video frames. DSTW has been designed to accommodate multiple hypotheses for the hand location in each frame. Therefore, we can afford to use a relatively simple and efficient hand detection scheme.

6.1 Hand Detection

We have performed hand detection in two steps: Skin color detection and Motion detection.

These visual cues used in our implementation for hand detection require only a few operations per pixel. Skin color detection involves only a histogram lookup per pixel (Jones and Rehg [18]) and is computationally efficient. Similarly, motion detection, which is based on frame differencing, involves a small number of operations per pixel. The motion detector computes a mask by thresholding the result of frame differencing (frame differencing is the operation of computing, for every pixel, the absolute value of the difference in intensity between the current frame and the previous frame). If there is significant motion between the previous and current frame the motion mask is applied to the skin likelihood image to obtain the hand likelihood image. We compute for every subwindow of some predetermined size the sum of pixel likelihoods(integral image) (Viola and Jones [19]) in that subwindow. Then we extract the K subwindows with the highest sum, such that none of the K subwindows may include the center of another of the K subwindows. If there is no significant motion between the previous and current frame, then the previous K subwindows are copied over to the current frame.

We have evaluated robustness to the size of the hand. We have taken hand sizes of 25 rows x 25 columns, size 35 rows x 35 columns, size 40 rows x 40 columns pixels. Size 25 rows x 25 columns pixels gave very poor result as it detected fingers due to small size of sub window. Example of window size 25 rows x 25 columns pixels is shown in Figure 6-1. Size 35 rows x 35 columns pixels gave good results for hand detection as the hand in our frames is roughly of that size without detecting other regions like face, knees, and individual fingers. So the K subwindows are constrained to the size 35 rows x 35 columns. Example is shown in Figure 6-3. Size 40 rows x 40 columns pixels gave good results but it detected additional components(knees, face) due to large size. Example of window size 40 rows x 40 columns pixels is shown in Figure 6-2. The number of K candidates is chosen by evaluating accuracy as shown in Figure 8-1.

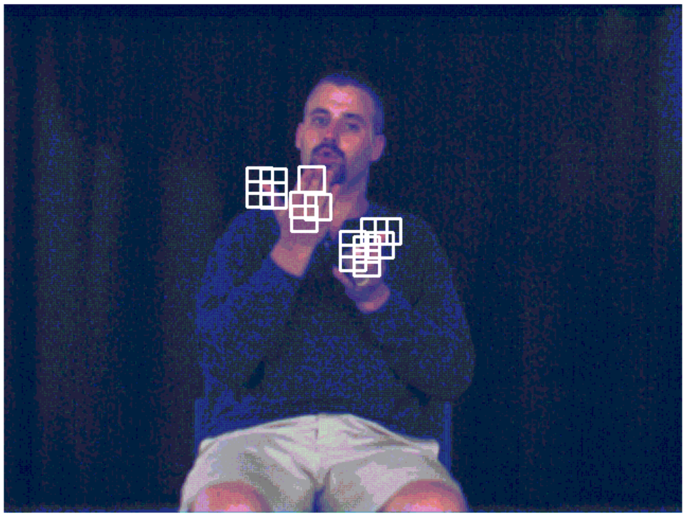


Figure 6-1 Subwindows size 25X25 of detected hand shown for a frame ‘t’ of a dataset video- observe the detected fingers

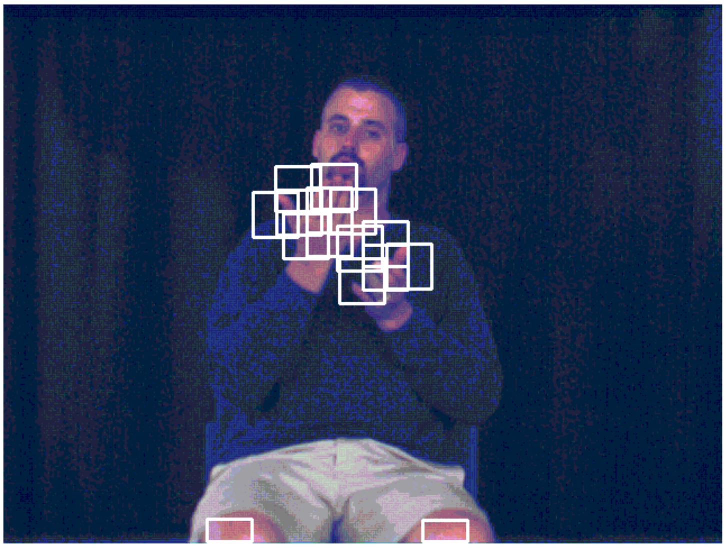


Figure 6-2 Subwindows size 40X40 of detected hand shown for a frame ‘t’ of a dataset video- observe the detected knee regions

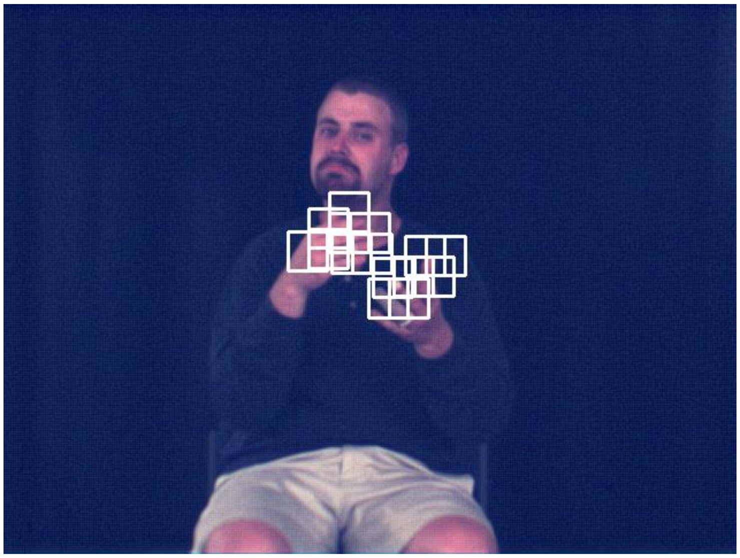


Figure 6-3 Subwindows of size 35X35 of detected hand shown for a frame ‘t’ of a dataset video

Our hand detection algorithm maintains for every frame of the sequence multiple subwindows, some of which may occupy different parts of the same connected component. The gesturing hand is typically covered by one or more of these subwindows.

6.2 Feature Extraction

Let X be a time series representation of database. We denote by |X| the number of frames in the video, and by X(t) the t-th frame of that video, t ranging from 1 to |X|. For the database video sequence, since we are given the position of the dominant hand in each frame, each sign video is naturally represented as a 2D time series ((x1, y1), …, (xn, yn)), where n is the number of frames in the video, and each (xi, yi) represents the pixel coordinates of the centroid of the hand in the i-th frame. We use notation M = (M1, …, Mm) for the model sequence, where each Mi is a feature vector (xi, yi).

Let Q be a time series representation of dataset video. For every frame j of the dataset sequence, the hand detector identifies K candidate hand regions. For every candidate k in frame j a 2D feature vector Qjk = (xjk, yjk) is extracted. The 2D position (x,y) is the region centroid.

Feature extraction for the dataset is done beforehand during the offline learning phase. For the digits dataset the offline feature extraction involves extracting features from hand locations detected when the user wears green colored gloves. Here the skin color detection phase is replaced by the color detection phase where the pixels which are green in color are obtained in the likelihood image by use of histograms. Using such additional constraints, like colored markers, is often desirable for the offline model building phase, because it simplifies the construction of accurate class models. It is important to stress that such markers are not used in the query sequences, and therefore they do not affect the naturalness and comfort of the user interface, as perceived by the end user of the system.

Feature extraction is a very phase in our gesture recognition process and it gives us the time series which can be input to the DSTW algorithm.

Chapter 7

Spatiotemporal Matching

In this chapter we describe the spatiotemporal matching algorithm that we have used. It can be applied in settings where neither spatial nor temporal segmentation of the input are available. To train the gesture models our first step is to learn the density functions, for each gesture class, using training examples for which spatiotemporal segmentation is known. For the Digits dataset green colored gloves are used for the training examples to facilitate automatic labeling of hand locations. Colored gloves are not used in the test sequences.

7.1 Dynamic Space Time Warping

After the model learning phase for the spatiotemporal matching we pass the features extracted from the Query and the Dataset to the Dynamic Space Time Warping(DSTW) Algorithm. DSTW is an extension of DTW that can handle multiple candidate detections in each frame of the dataset. A key limitation of DTW when applied to gesture recognition is that DTW requires knowledge of the location of the gesturing hands in both the database videos and the query video. In contrast, DSTW only requires known hand locations for the database videos.[6] Given a dataset video, instead of requiring the hand locations to be specified (as DTW does), DSTW makes the much milder assumption that a hand detection module has produced a relatively short list of candidate hand locations. The ability to handle multiple candidate hand locations allows DSTW to be seamlessly integrated with existing imperfect hand detectors. As a result, DSTW-based systems can be readily deployed for gesture recognition in challenging real-world environments.

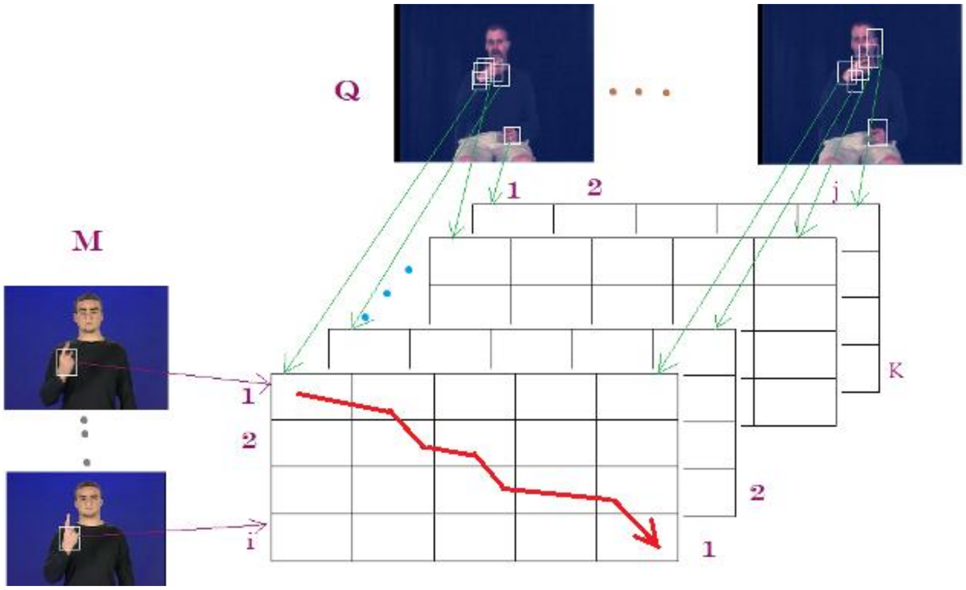


Figure 7-1 Dynamic Space Time Warping

7.2 DSTW Algorithm & Warping Path

For the query video sequence, since we are given the position of the dominant hand in each frame, each sign video is naturally represented as a 2D time series ((x_1, y_1), …, (x_n, y_n)), where n is the number of frames in the video, and each (x_i, y_i) represents the pixel coordinates of the centroid of the hand in the i-th frame. We use notation M = (M_1, …, M_m) for the model sequence, where each M_i is a feature vector (x_i, y_i). Let Q = (Q_1, …, Q_n) be for dataset sequence. In the regular DTW framework, each Q_j would be a feature vector, of the same form as each M_i. However, in dynamic space-time warping (DSTW), we want to model the fact that we have multiple candidate feature vectors in each frame of the dataset. For example, if the feature vector consists of the position and velocity of the hand in each frame, and we have multiple hypotheses for hand location, each of those hypotheses defines a different feature vectorIn DSTW, Q j is a set of feature vectors: Q_j = {Q_{j1}, …, Q_{jK}}, where each Q_{jk}, for k = {1, …, K}, is a candidate feature vector. K is the number of feature vectors extracted from each dataset frame. In our algorithm we assume K is fixed, but in principle K may vary from frame to frame.

A warping path W in DSTW defines an alignment between M and Q. Each element of W is a triple: W=(($w_{1,1}$, $w_{1,2}$, $w_{1,3}$), …, ($w_{|W|,1}$, $w_{|W|,2}$, $w_{|W|,3}$)). Triple ($w_{i,1}$, $w_{i,2}$, $w_{i,3}$) specifies a correspondence between frame $w_{i,1}$ of Q and frame $w_{i,2}$ of X, but also specifies that, out of the multiple candidate hand locations in the $w_{i,1}$-th frame of Q, the candidate hand location indexed by $w_{i,3}$ is the one that optimizes the similarity score between dataset and model sequence.

7.3 Legal Warping Path

We say that w_t has two temporal dimensions (denoted by i and j) and one spatial dimension (denoted by k). The warping path is typically subject to several constraints (adapted from [11] to fit the DSTW framework):

• **Boundary conditions:** w_1= (1,1, k) and w_T=(m, n, k). This requires the warping path to start by matching the first frame of the model with the first frame of the query, and end by matching the last frame of the model with the last frame of the query. No restrictions are placed on k and k, which can take any value from 1 to K.

• **Temporal continuity:** Given w_t = (a, b, k) then w_t−1 = (a, b , k), where a − a≤ 1 and b − b≤ 1. This restricts the allowable steps in the warping path to adjacent cells along the two temporal dimensions.

• **Temporal monotonicity:** Given w_t = (a, b, k) then w_t−1= (a, b , k) where a − a≥ 0 and b − b≥0. This forces the warping path sequence to increase monotonically in the two temporal dimensions.

Note that continuity and monotonicity are required only in the temporal dimensions. No such restrictions are needed for the spatial dimension; the warping path can “jump” from any spatial candidate k to any other spatial candidate k. Given warping path element w_t= (i, j, k), we define the set N(i, j, k)to be the set of all possible values of wt−1 that satisfy the warping path constraints (in particular continuity and monotonicity): N(i, j, k) = {(i−1, j),(i, j−1),(i−1, j−1)}×{1,...,K}

We assume that we have a cost measure d(i, j, k) ≡d(M_i, Q_{jk}) between two feature vectors M_i and Q_{jk}.

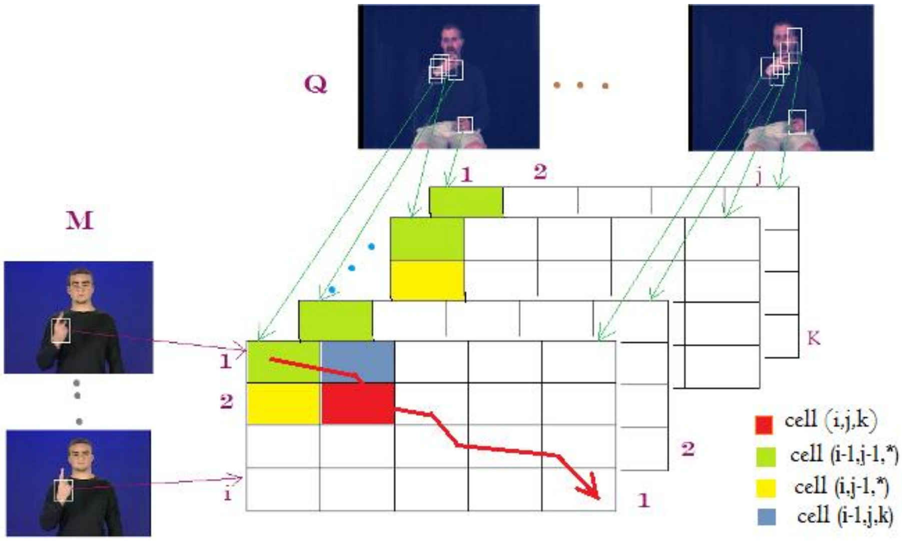
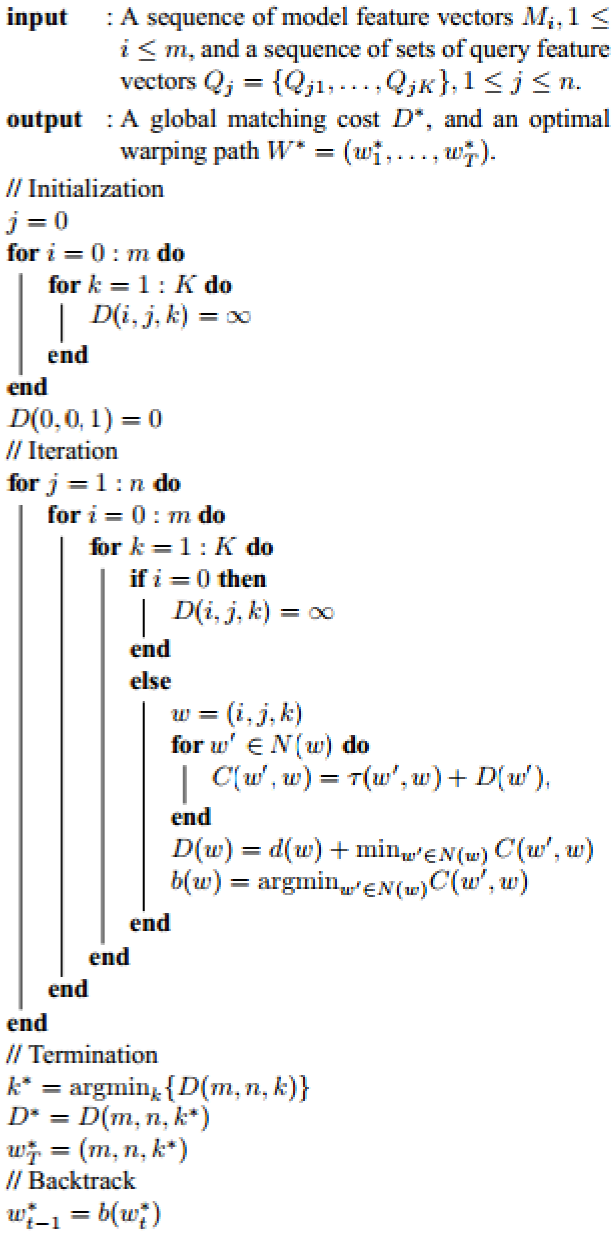


Figure 7-2 DSTW- Break problem up into smaller, interrelated problems(i,j,k)



```
input   : A sequence of model feature vectors $M_i, 1 \leq i \leq m$, and a sequence of sets of query feature
          vectors $Q_j = \{Q_{j1}, \ldots, Q_{jK}\}, 1 \leq j \leq n$.
output  : A global matching cost $D^*$, and an optimal warping path $W^* = (w_1^*, \ldots, w_T^*)$.
// Initialization
$j = 0$
for $i = 0 : m$ do
    for $k = 1 : K$ do
        $D(i, j, k) = \infty$
    end
end
$D(0, 0, 1) = 0$
// Iteration
for $j = 1 : n$ do
    for $i = 0 : m$ do
        for $k = 1 : K$ do
            if $i = 0$ then
                $D(i, j, k) = \infty$
            end
            else
                $w = (i, j, k)$
                for $w' \in N(w)$ do
                    $C(w', w) = \tau(w', w) + D(w')$,
                end
                $D(w) = d(w) + \min_{w' \in N(w)} C(w', w)$
                $b(w) = \text{argmin}_{w' \in N(w)} C(w', w)$
            end
        end
    end
end
// Termination
$k^* = \text{argmin}_k \{D(m, n, k)\}$
$D^* = D(m, n, k^*)$
$w_T^* = (m, n, k^*)$
// Backtrack
$w_{t-1}^* = b(w_t^*)$
```

Figure 7-3 The DSTW Algorithm [7]

7.4 Incorporating transition costs

The hand in one frame should not be too far and should not look too different from the hand in the previous frame. We assume from [7] that we have transition cost T (wt-1; wt) between two successive warping path elements. DSTW finds the optimal path W* and the global matching score D* as described in Algorithm 1. For this algorithm to function correctly it is required that T (w', w) = 0 when w'= (0, j, k) or w'=0 = (i, 0, k). For all other values of w', T must be defined appropriately in a domain-specific way. The function T plays a similar role in DSTW as state transition probabilities play in the HMM framework.

7.5 Comparing Trajectories via DSTW

The cost C(W) of warping path W that we use is the sum of the Euclidean distances between the |W| pairs of corresponding feature vectors defined by W. In matching Q with M, the number of possible warping paths for DSTW is exponential to |Q| and to |M|. Despite the exponential number of possible warping paths for DSTW, the optimal warping path can still be found efficiently, in polynomial time, using dynamic programming. The DSTW distance between Q and M is defined as the cost of the optimal (min-cost) warping path between Q and M. DSTW finds the optimal path W∗ and the global matching score D∗. After we have computed all values D(i, n, k), the minimum cost among all D(m, n, k) (i.e., all costs corresponding to a match between the last state of the model and the current frame n) corresponds to the optimal alignment between the model and the video sequence. Using backtracking, i.e., following predecessor links, we obtain the optimal warping path. DSTW takes O(Kmn) time.

Chapter 8

Experiments

The DSTW Algorithm for gesture spotting is evaluated using the Digits Dataset and the ASL Narratives Dataset. We have implemented the hand-signed recognition system in Java integrated with the user interface in Matlab.

8.1 Calculating Accuracy: Symmetric Distance Algorithm

The symmetric distance algorithm to calculate accuracy to know whether gesture is correct or not is as follows:

When a system decides that a gesture occurred at frames (A1,…B1) we consider the results correct when:

- There was some true gesture at frames (A2,….,B2)
- At At least half+1 (overlap range) of the frames in (A2, …, B2) are covered by (A1, …, B1)
- At least half+1 (overlap range) of the frames in (A1, …, B1) are covered by (A2, …, B2)
- The class of the gesture at (A2 B2) matches the – The class of the gesture at (A2, …, B2) matches the class that the system reports for (A1, …, B1)

The examples in Figures 8-1 and 8-2 demonstrate how the symmetric distance algorithm works in determining whether the gesture is classified accurately or not. In Figure 8-1 the system decides that the gesture sign is spotted correctly by the algorithm after following all the steps in the symmetric distance algorithm as mentioned above. In Figure 8-2 the system decides that the gesture sign is spotted incorrectly by the algorithm after following all the steps in the symmetric distance algorithm as mentioned above.

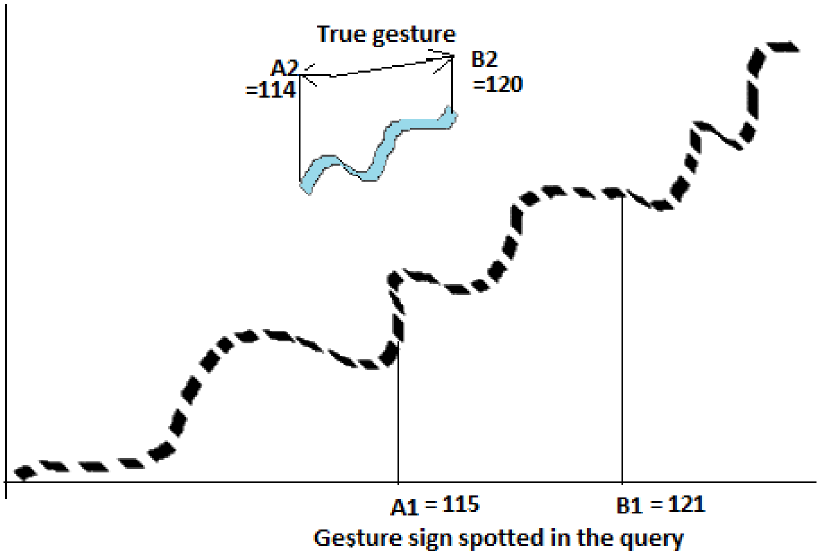


Figure 8-1 Example of Symmetric Distance Algorithm classifies the gesture as correctly spotted.

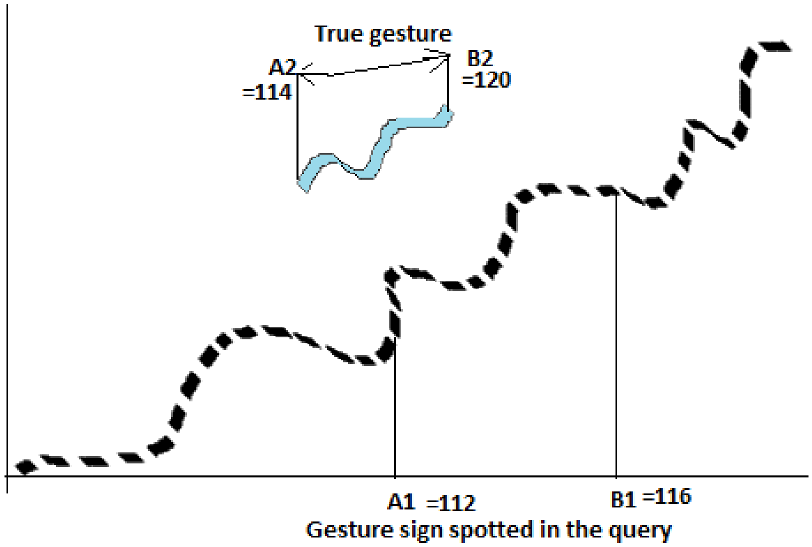


Figure 8-2 Example of Symmetric Distance Algorithm classifies the gesture as incorrectly spotted.

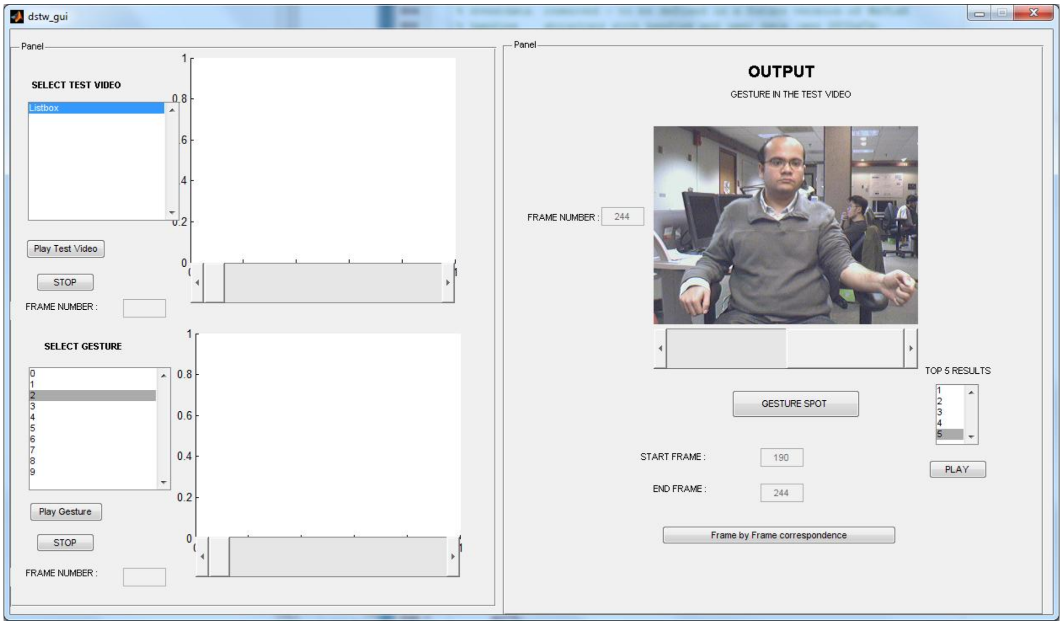
We have observed classification accuracy for different ranges for the symmetric distance algorithm. We have tried ranges R= {half+1, onethird+1, one fouth+1}. We have used the range half+1 henceforth in our results. One third +1 works similarly giving few more matches to half+1 results without increasing redundant matches. Onefourth+1 gives a lot more redundant matches and is does not provide very good results.

8.2 Performance Evaluation on Digits Dataset

We have rebuilt a system from [5] to evaluate the performance of DSTW on the digits dataset. For the experiments we used video clips of three users gesturing the ten digits in the style of Palm's Grafitti Alphabet(Figure 3.1). The video clips were captured with a Logitech 3000 Pro camera using an image resolution of 240 × 320. A total of 270 digit exemplars were extracted from three different types of video clips depending on what the user wore:

- Colored Gloves: 30 digit exemplars per user used as training dataset.
- Long Sleeves: 30 digit exemplars per user were used as queries.
- Short Sleeves: 30 digit exemplars per user were used as queries.

Given a query frame, K candidate hand regions of size 25 X 25 were detected as described in Section 5.1. For every candidate hand region in every query frame, a feature vector was extracted and normalized as described in Section 5.2. For any digit gesture, the feature vector extracted for the query sequence will be the hand trajectory matrix of entire query video (From the Easy or Hard test video set of digits dataset) and training dataset sequence would be the green hand trajectory of the digit gesture from each of 30 colored gloves training sequence so there will be total 30 green hand trajectories as dataset sequence input to DSTW.



FRAME NUMBER : 244

START FRAME : 190

END FRAME : 244

Figure 8-3 GUI for the dynamic gesture spotting algorithm used on Digits Dataset. It display top 5 results, the start and the end frames retrieved after running the algorithm and has a button for frame-by-frame correspondence(see figure 8-2)

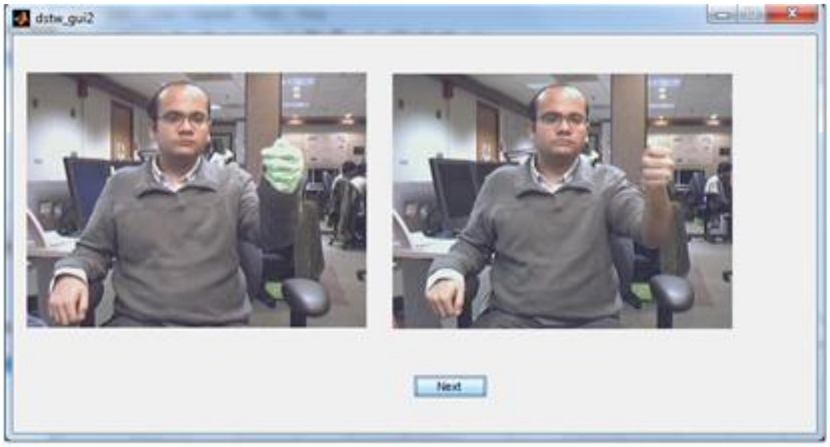


Figure 8-4 Frame by Frame correspondence button (in Figure 8-1) opens a new GUI window which displays images according to the DSTW alignment between query video and the dataset.

Classification of a gesture G in any digit class (0-9) is done in the following steps:

- Compute DSTW score between input gesture G and each M*N training examples. (N classes, M examples per class)

- We keep track, for each of the M*N examples, of the dynamic programming table constructed by matching those examples with Q1, Q2 ,….Qt-1.

- At frame t, we add a new column to the table.

- If, for any training example, the matching cost is below a threshold we "recognize" a gesture.

Using the Symmetric distance algorithm mentioned in Section 8.1 we evaluate the accuracy of the gesture recognition system on the digits dataset. We have chosen different values of K and checked the accuracy. From Table 8.1 we can see that K=15 gives the best results.

Table 8-1 Measuring accuracy with different values of K candidates

K	Correct out of 20 (short sleeve gestures)	Accuracy(%)
7	14	70
10	16	80
12	15	75
13	15	75
15	17	90

The results are as follows:

- Easy test set: 256 gestures correctly recognized out of 300. Accuracy of the algorithm on easy test set ≈ 85.33%.

- Hard Test set: 83 gestures correctly recognized out of 140. Accuracy of the algorithm on hard test set ≈ 59.28%

8.3 Cross Platform Implementation- Matlab with Java

We have built our ASL gesture spotting system using Matlab integrated with Java. We have implemented the DSTW algorithm in Java using Eclipse tool and then used the Java class in Matlab.

Input variables to Java DSTW algorithm:

Query array: A 2D array of type Double. It consists of the hand trajectory matrix. It gives positions of hand co-ordinates for each frame of the training sequence.

Dataset array: A 3D array of type Double. It consists of the hand trajectory matrix. It consists of best K candidate hand positions of hand co-ordinates for each frame of the test sequence.

Number of frames in Query sequence: It is an integer that gives this number.

Number of frames in Dataset sequence: It is an integer that gives this number.

Output Variables:

Array D: A 3D array of warping paths which has values calculated for D which is then used in Matlab to find end frame of the optimal match and then to backtrack to start frame.

Accuracy:

It gives exactly same accuracy as the algorithm without using Java and only in Matlab.

Runtimes:

For the Digits Dataset

Without using Java: 4.9407e+03 sec

Using Java: 196.26 secs

The program is run for 1 test video with all 10 gestures and 30 training videos per gesture. There are 10 gestures so we have a total of 300 training sets.

Total time to recognize start and end frame for all 10 gestures for 1 video with all 300 training~=197.26 sec

Time to get hand trajectory array (k=15) for the test video=109.47sec

Time to compare test video against only 1 gesture (30 training samples of same gesture) =10.787sec

Time to compare test video against 1 training sample of a gesture=

Time to find end (0.3765) (within this the java class uses time 0.2765 to generate matrix “D”) + time to backtrack to the start(0.0013389)

If we eliminate the time to get hand trajectory array for test video then time would be=197.26-109.47= 87.79secs.

~96% decrease in time compared to the Matlab program.

For ASL Narratives Dataset:

Time to get hand trajectory array for the dataset video for a single story~=7240 This is the time consuming phase and is done as a part of preprocessing.

Time to compare one Query video with the entire dataset(combination of all 12stories) using cross platform DSTW spotting algorithm ~= 2sec (excludes time to get hand trajectory)

8.4 Performance Evaluation on ASL Narratives Dataset

The system is said to succeed if it can find the correct start and end frame of the occurrences for the query in the dataset. The ASL Narratives dataset is different from the Digits Dataset in several ways:

- The training dataset sequence in the digits dataset had Green Hand gestures used for training. Here dictionary has only normal hand gestures so we use only skin detection in the offline learning phase.

- Single video is used for offline training per gesture sign. This video is obtained from the ASLLVD[1] and has sign from the Gallaudet dictionary.

- The dataset is an one entire video from ASL narrative story video dataset described in section 3.2.

Our gesture spotting algorithm in evaluated only on one-handed signs of ASL. The algorithm basically focuses on movement of the hand and the location of the hand in the frame. The scores obtained from the DSTW algorithm are used to find the top ‘N’ matching results. Classification accuracy is obtained using the symmetric distance algorithm explained in section 8.1. Other evaluation measures used are as follows:

- Precision is the fraction of retrieved instances that are relevant

- Recall (Retrival ratio) is the fraction of relevant instances that are retrieved.

To avoid multiple results pointing to the same frame we have suppressed such results by using a threshold so that if the next found result is very close to the first one then it is discarded. (the best result was obtained with threshold=15)

The average lengths of dataset videos are shown in Table 3.1. The Evaluation Measure used is shown below:

- N = Number of top ‘N’ results that we want as output

For each N:

Get top N results

Calculate Precision: Percentage of correct results

Calculate Recall: Percentage of correct results that are in top N

- Find the Rank of the highest-ranking correct match.

Protocol used to select gallaudet dictionary dataset sign:

- Found that our ASL Narratives story dataset has 734 different signs from Gallaudet dictionary.
- Out of those 160 signs are one-handed signs
- Algorithm is run on 60 random signs from the 160 one-handed signs that occur at least once in the entire dataset of stories and on ASL story dataset.

We have measured rank of the highest ranking correct, precision and recall (retrieval ratio) for Top 50, 100, 200 and 300 results for the gestures with and without transition costs. Graph in Figure 8-5 shows the Precision-Recall relation for DSTW gesture spotting algorithm with transition costs when the algorithm is run on 60 random signs from the 160 one-handed signs that occur at least once in the entire dataset of stories and on ASL story dataset. Average precision and recall is calculated for the 60 queries at 50,100,200 and 300 top results. Here x={R1,R2,R3,R4} where R1 is average recall for 60 queries for top 50 results, R2 is average recall for 60 queries for top 100 results, R3 is average recall for 60 queries for top 200 results and R4 is average recall for 60 queries for top 50 results. Here y={P1,P2,P3,P4} where P1 is average precision for 60 queries for top 50 results, P2 is average precision for 60 queries for top 100 results, P3 is average precision for 60 queries for top 200 results and P4 is average precision for 60 queries for top 50 results. The error E is obtained using the formula E = std(x)*ones(size(y)) in Matlab where the function std() gives standard deviation. E is represented using error bars in the graph of figure 8-5.

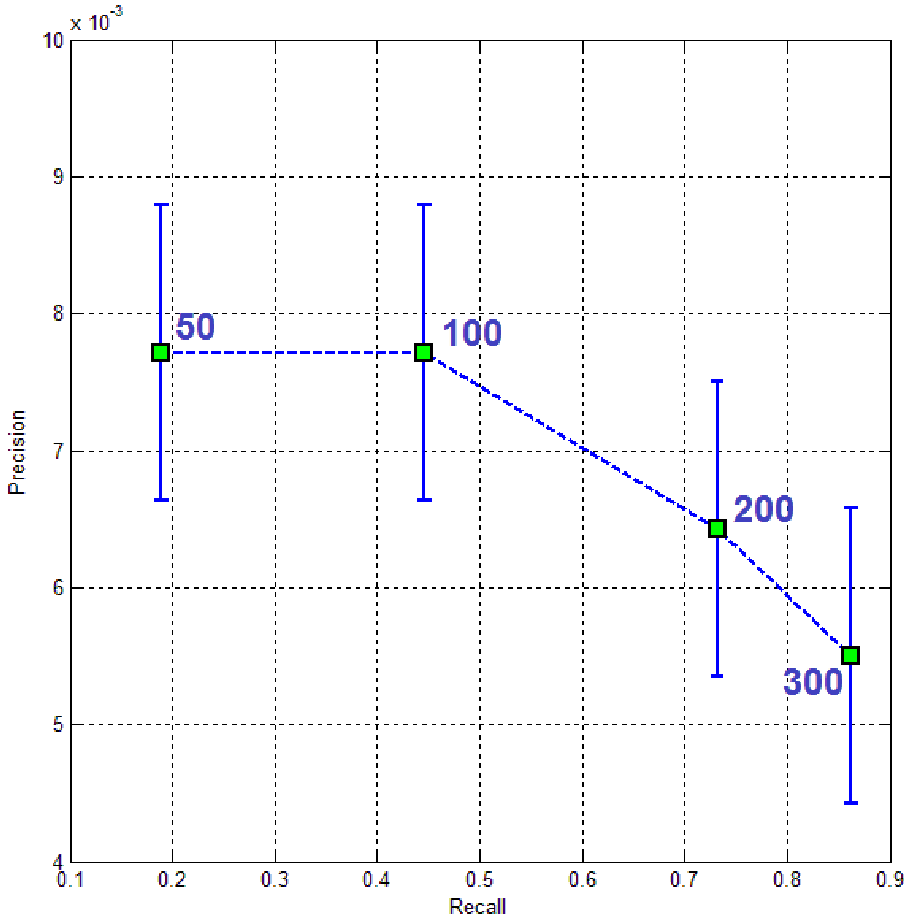


Figure 8-5 Graph to plot Precision-Recall relation for average over the 60 queries at 50,100, 200 and 300 top results of DSTW gesture spotting algorithm with transition costs.

Graph in Figure 8-6 shows the results on one-handed gestures for 60 signs on complete story dataset (combining all stories). It plots the rank of the highest ranking correct match on x axis and shows the percentile of results corresponding to that rank on the y co-ordinate.

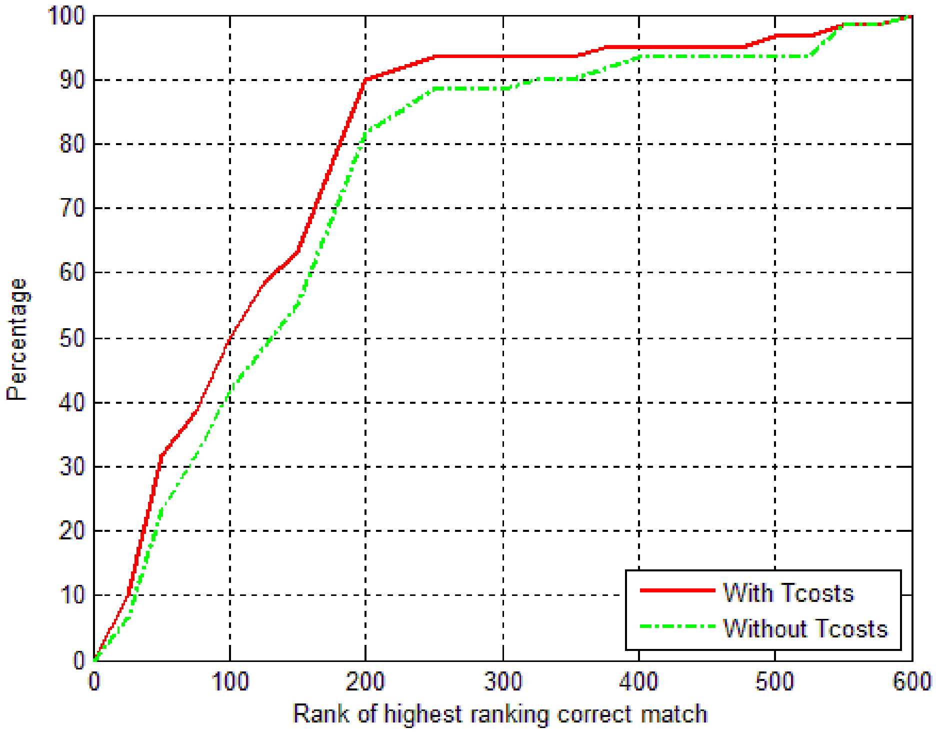


Figure 8-6 Graph to plot Rank of Highest Ranking correct match Vs. Percent of Results for 60 signs on complete story dataset (combining all stories)

We have analyzed what works well and what does not for improving accuracy of the DSTW gesture spotting algorithm. We have done analysis of motion in an ASL sign and compared results for signs which exhibit more motion compared to the signs which exhibit the least motion. Graph in Figure 8-7 shows the results on one-handed gestures on story videos for 20 signs which exhibit the most motion and for 20 signs which exhibit the least motion. We have evaluated the performance with and without using transition costs. The graph plots the rank of the highest ranking correct match on x axis and shows the percentile of results corresponding to that rank on the y co-ordinate.

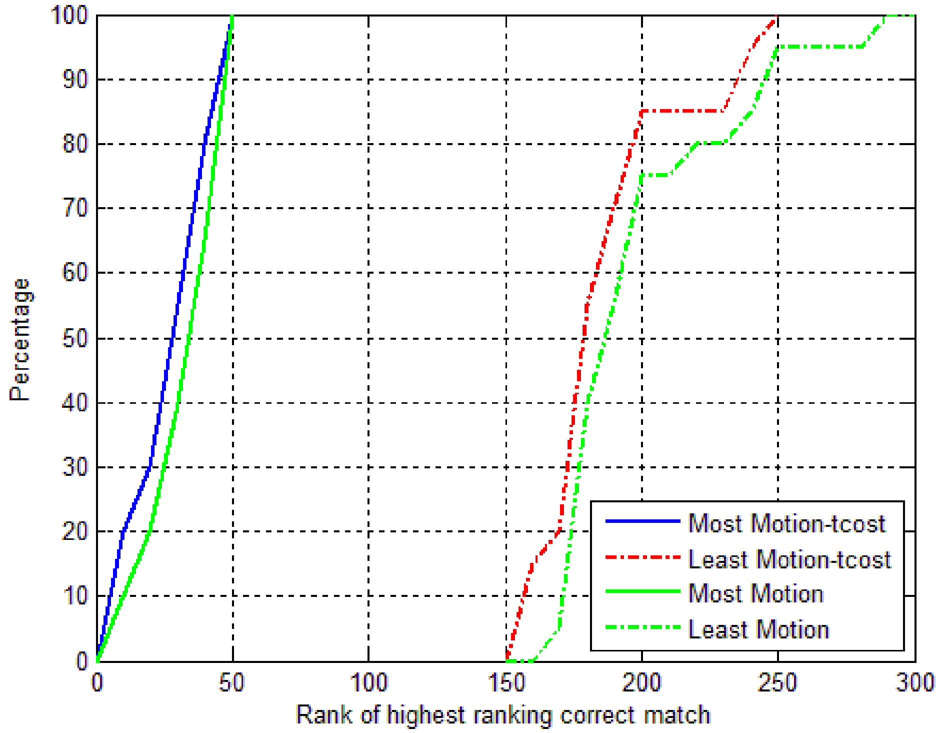


Figure 8-7 Graph to plot Highest Ranking correct match Vs. Percent of Results videos for 20 signs which exhibit the most motion and for 20 signs which exhibit the least motion(with and without tcosts).

Figures 8-8 to 8-10 show example frames for few ASL signs which exhibit more motion and give good results. Figures 8-11 to 8-13 show example frames for few ASL signs which exhibit less motion and do not give very good results. The figures show a frame from the beginning, the middle and the end of the sign.

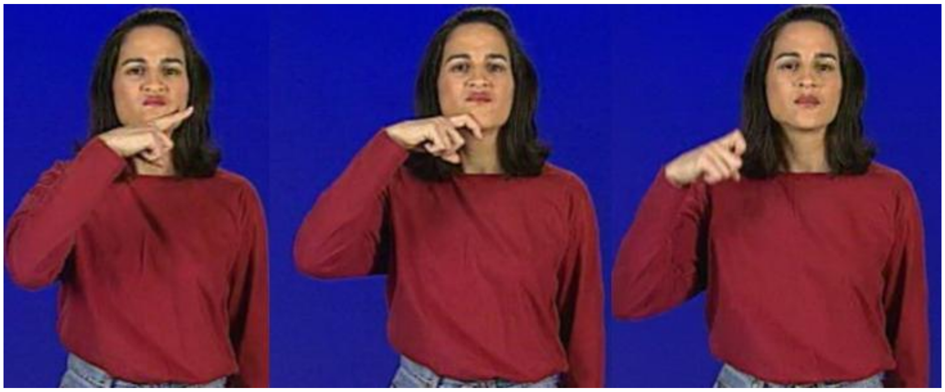


Figure 8-8 ASL Sign for “Dry” exhibits more motion.

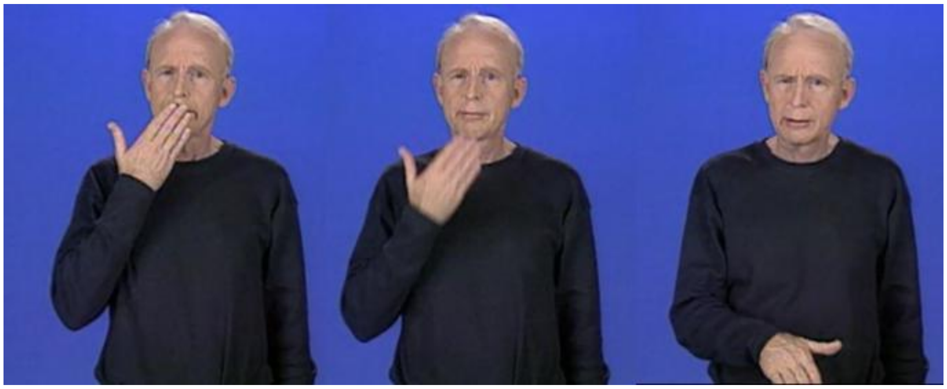


Figure 8-9 ASL Sign for “Bad” exhibits more motion.

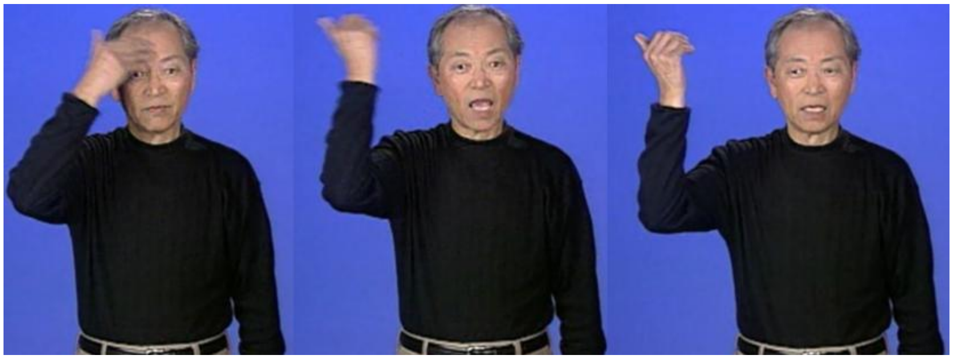


Figure 8-10 ASL Sign for “New” exhibits more motion.

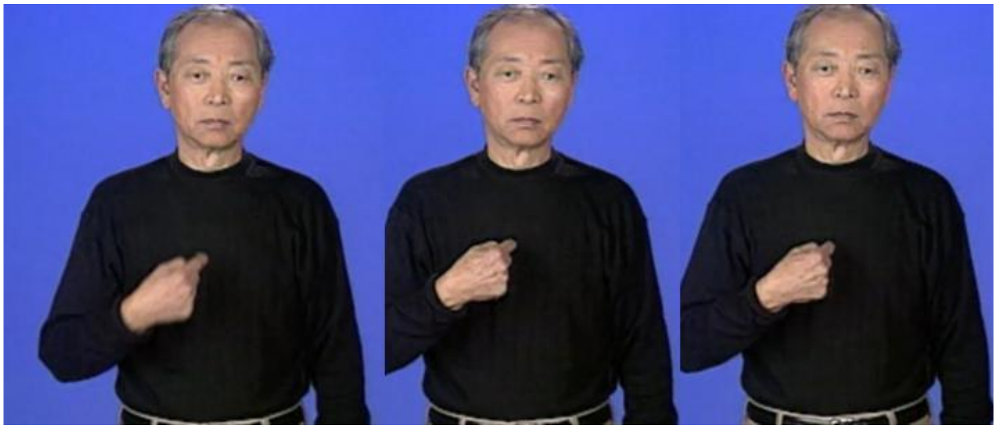


Figure 8-11 ASL Sign for “Me” exhibits less motion.

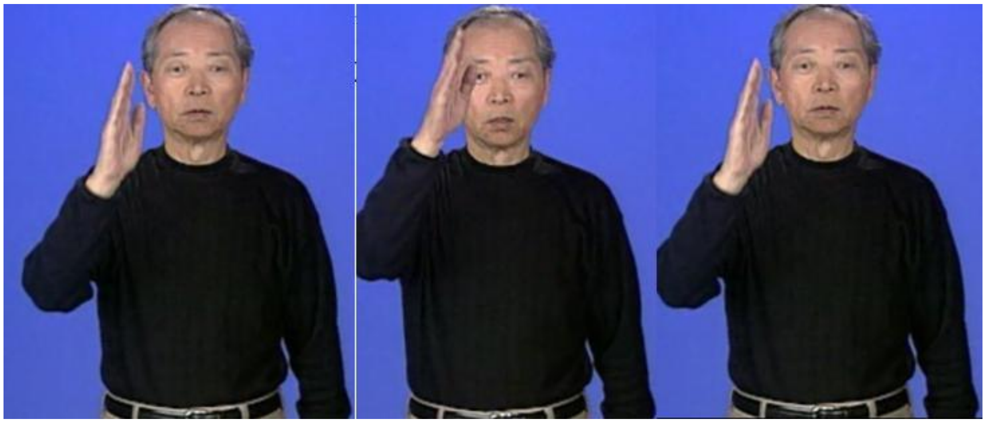


Figure 8-12 ASL Sign for “God” exhibits less motion.

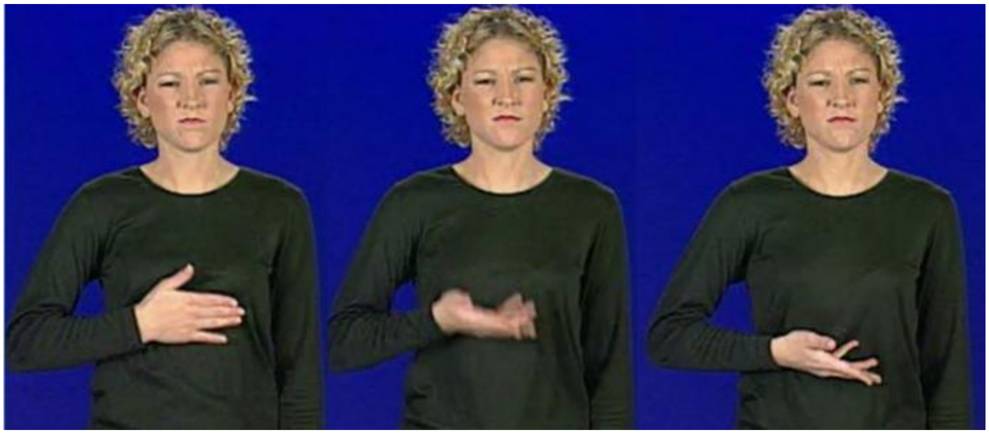


Figure 8-13 ASL Sign for “Upset” exhibits less motion.

Chapter 9

Discussion & Future work

We have shown a method for helping users look up unknown signs from an ASL story/ narrative containing large number of signs, using similarity-based DSTW retrieval method. We have defined feature vectors based on hand motion and hand appearance. Similarity between signs is measured by dynamic time warping scores, which are based on hand motion. There are several research topics that we are interested in pursuing as futurework, with the goal of further improving system performance and the overall user experience.

Our gesture recognition algorithm basically focuses on movement of the hand and the location of the hand in the frame. We observed that many one handed signs in ASL that don't use a lot of movement. For example the ASL gesture sign for 'COP' is making a C shape with the dominant hand and placing on the opposite shoulder and holding it for a couple of seconds. The dataset had the exact gesture which is easily visible to the eye but the algorithm is not able to comprehend it as it only uses the trajectory of the hand as the feature and not the shape that it forms. Whether the hand forms a fist or C shape does not make any difference to the algorithm. It would be a research topic to try and incorporate finger detection and movement for gesture recognition. Another example is when the algorithm was run to find 'I/ME' it showed location dataset lapd_1 which actually had gesture 'THUMB' pointing behind. This was fairly similar to the gesture 'I/ME'. The only difference between the two gestures is 'I/ME' uses the index finger and performs the same movement and 'THUMB' uses the thumb and performs the same gesture. Figure 9-1 shows these gestures and their similarities.

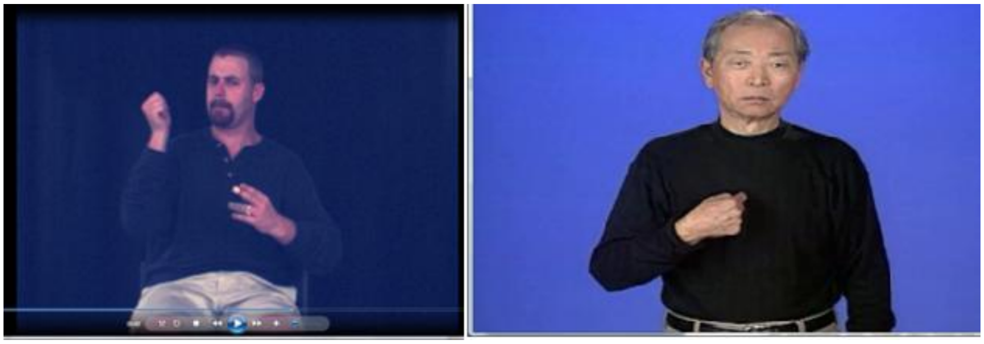


Figure 9-1 Gesture ‘I/ME’ (right) recognized as gesture ‘THUMB’ (left)

Finger detection could bring a very big improvement to the system as many ASL signs are subtly different and need observation of finger movements and not just the entire hand.

Another improvement that can be made is in some way to incorporate depth into the algorithm. For example for the sign for ‘SEE’ the person moves two fingers from his eyes to a distance away at the same height of the frame. So here the hand only gets closer to the camera. But the algorithm considers only the center for the hand and that does not change on it coming closer. So here depth matters and not the location and movement.

The gesture spotting algorithm is efficient, purely vision-based, and can robustly recognize gestures, even when the user performs gestures without the help of any aiding devices and with a complex background. The proposed approach works reasonably well in our experiments and we believe that more work is needed in order improve it further and find a good similarity measure that incorporates finger detection and depth and can be used for a large vocabulary of signs, given only one or two training examples per sign.

References

[1] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, “The American Sign Language Lexicon Video Dataset," *in IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB),* 2008.

[2] R. A. Tennant and M. G. Brown, *The American Sign Language Handshape Dictionary*. Gallaudet U. Press.

[3] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, , and F.Kamangar., “A System for Large Vocabulary Sign Search," *in Workshop on Sign, Gesture and Activity (SGA)*, September 2010, pp. 1-12.

[4] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, A. Thangali, H. Wang, and Q. Yuan, “Large Lexicon Project: American Sign Language Video Corpus and Sign Language Indexing/Retrieval Algorithms," *in 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign LanguageTechnologies*, 2010, pp. 1-4.

[5] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff, “A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 31(9),* pages 1685-1699, September 2009.

[6] Haijing Wang, Alexandra Stefan, and Vassilis Athitsos, “A Similarity Measure for Vision-Based Sign Recognition,” *Invited submission to International Conference on Universal Access in Human-Computer Interaction (UAHCI)*, pages 607-616, July 2009.

[7] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff, “Simultaneous Localization and Recognition of Dynamic Hand Gestures,” *IEEE Motion Workshop*, pages 254-260, January 2005.

[8] Eamonn Keogh and Chotirat Ann Ratanamahatana. “Exact indexing of dynamic time warping,” *Springer-Verlag London Ltd*., 2004

[9] A. Corradini, “Dynamic time warping for off-line recognition of a small gesture vocabulary,” *in Proc. IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 2001, pp. 82–89.

[10] Kruskal, J.B., Liberman, M., “The symmetric time warping algorithm: From continuous to discrete,” *in Time Warps. Addison-Wesley*, 1983

[11] Michalis Potamias and Vassilis Athitsos, “Nearest Neighbor Search Methods for Handshape Recognition,” *Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, July 2008.

[12] Zhong Zhang, Rommel Alonzo, and Vassilis Athitsos, “Experiments with Computer Vision Methods for Hand Detection,” Conference on Pervasive Technologies Related to Assistive Environments (PETRA), May 2011.

[13] Q. Yuan, S. Sclaroff and V. Athitsos, “Automatic 2D hand tracking in video sequences,” *in IEEE Workshop on Applications of Computer Vision*, pp. 250–256, 2005.

[14] Sergio Escalera, Alberto Escudero, Petia Radeva, and Jordi Vitria, “Adaptive Dynamic Space Time Warping for Real Time Sign Language Recognition,” *Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain UB Dept. Matematica Aplicada i An ` alisi, UB, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain.*

[15] Stan Salvador and Philip Chan, “Toward Accurate Dynamic Time Warping in Linear Time and Space,” *Work performed at the Florida Institute of Technology.*

[16] Manavender R. Malgireddy, Jason J. Corso and Srirangaraj Setlur, “A Framework for Hand Gesture Recognition and Spotting Using Sub-gesture Modeling,” *in International Conference on Pattern Recognition*, 2010.

[17] Thomas Stiefmeier, Daniel Roggen and Gerhard Troster, “Gestures are Strings: Efficient Online Gesture Spotting and Classification using String Matching,” *work at Wearable Computing Lab, ETH Zürich, Zurich, Switzerland*

[18] M. Jones and J. Rehg,n, “Statistical color models with application to skin detection,” *Intenational Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, January 2002.

[19] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *in Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. I:511–518.

[20] Vassilis Athitsos, Haijing Wang, Alexandra Stefan, “A Database-Based Framework for Gesture Recognition,” Journal of Personal and Ubiquitous Computing, 14(6), pp 511-526, September 2010.

[21] Alexandra Stefan, Vassilis Athitsos, Jonathan Alon, and Stan Sclaroff, “Translation and Scale-Invariant Gesture Recognition in Complex Scenes,” *Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, July 2008.

[22] Stan Sclaroff, Margrit Betke, George Kollios, Jonathan Alon, Vassilis Athitsos, Rui Li, John Magee, and Tai-peng Tian, “Tracking, Analysis, and Recognition of Human Gestures in Video,” *Invited submission to International Conference on Document Analysis and Recognition (ICDAR)*, pages 806-810, August 2005.

[23] Alexandra Stefan, Haijing Wang, and Vassilis Athitsos, “Towards Automated Large Vocabulary Gesture Search,” *Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, June 2009.

[24] Paul Doliotis, Alexandra Stefan, Chris Mcmurrough, David Eckhard, and Vassilis Athitsos. “Comparing Gesture Recognition Accuracy Using Color and Depth Information,” *Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, May 2011.

Biographical Information

Srujana Gattupalli was born in Maharastra, India in 1989. She has received her B.E in Computer Engineering from Gujarat University, India, in 2011, her M.S. degree in Computer Science from The University of Texas at Arlington in 2013. She has worked as a software developer intern for the development of Enterprise Resource Planning project at Additol Lubricants Ltd. Her current research interests are in the area of Computer Vision, Data mining, Data Simulation & Modeling and Image processing.