IMPROVING ACCURACY IN LARGE VOCABULARY SIGN SEARCH SYSTEMS

by

CHRISTOPHER CONLY

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

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2016

To my mother, Judy, and father, John, for their support and encouragement over the years and to my siblings Carol, James, and Elizabeth, without whom, I would not be today where I am or the person I am. I thank and love you all.

ACKNOWLEDGEMENTS

I would first like to thank Vassilis Athitsos—my advisor, my mentor, my reality check, and my friend—for his sound guidance, his steadfast support, his patience and understanding, his generosity, his encouragement when I needed it most, and his belief in me when I doubted myself. Vassilis, I offer you my most sincere gratitude. I would also like to thank the members of my committee, Farhad Kamangar, Heng Huang, and David Levine, as well as Gian Luca Mariottini for their input and guidance over the years and for being inspiring teachers who always pushed me to learn. I enjoyed working with you all and look forward to continued relationships.

I also thank everyone in the VLM lab who helped me put the system through its paces. Alex Dillhoff, Chris McMurrough, Zhong Zhang, Paul Doliotis, Dave Cavaletto, Claire Henry, Wei Xiang, and Pat Jangyodsuk for being good sounding boards for ideas. Our discussions over the years have been very helpful. Thank you Srujana Gattupalli for always being there to help me with demos and lab tours; I really appreciate it. I also thank the rest of the VLM crew–Amir Ghaderi, Sakher Ghanem–and Matt Middleton for participation in my experiments. I look forward to working further with you all.

Without the help of Joan Bempong and Carolyn Stem, we never would have been able to record the ASL dataset. Fadia al Qudah and Sylvia Loh were instrumental in providing enough annotations to actually do some experiments. I am also grateful for Zachary Staranowicz and Ricky Stafford for agreeing to let us record them as signers and for helping with the painstaking annotations.

May 23, 2016

ABSTRACT

IMPROVING ACCURACY IN LARGE VOCABULARY
SIGN SEARCH SYSTEMS

Christopher Conly, Ph.D.

The University of Texas at Arlington, 2016

Supervising Professor: Vassilis Athitsos

An automated computer vision-based dictionary system for looking up the meaning of signs can be an invaluable tool both for students and native users of sign languages. Students may not know the meaning of a sign they encounter and would like to learn what it is. A native signer knows what it means to them but may be unsure of the equivalent in English. Such a system can return a ranked video list of the most similar signs to a query video and allow the user to browse the video results to find the desired sign and its meaning. This thesis investigates and proposes improvements in large vocabulary sign search systems and culminates in an automated American Sign Language dictionary search system with improved accuracy over former variants.

This type of dictionary system presents several challenges. When a large vocabulary is desired, it is often not feasible to generate a large enough training set to train statistical and machine learning recognition methods that have achieved good accuracy on smaller vocabularies. In this case, exemplar-based methods must be used and improved upon. Secondly, there are large variations in the performance of signs

inherent in user-independent systems. Generative statistical methods like Hidden Markov Models can model these variations but may be unusable in such a system due to the insufficient number of training samples required for learning transition probabilities.

This thesis makes the following contributions. First, there is a lack of publicly available, fully annotated, large vocabulary RGB-D gesture datasets for use in gesture recognition research. Thus, a multimodal 3D body part detection and large vocabulary American Sign Language dataset is presented that allows researchers to evaluate both body part (i.e. hands and shoulders) detection and gesture recognition methods. This dataset is used to establish benchmarks and for testing the methods developed in this work. The primary differences between this dataset and others are the vocabulary size and the full annotations of joint positions in every frame of each gesture.

Second, this thesis proposes *Intra-Class Variation Modeling*, a method that addresses the wide variability in sign performance by generating models for same-class differences in several geometric properties of the hand trajectories comprising the signs. These models can be used to generate features that describe the likelihood that a query sign matches an example sign given the observed differences in these properties and provide an improvement to the exemplar-based similarity measure.

The third contribution of this work is *Multiple-Pass Dynamic Time Warping*, a way to better handle various size and spatial translation differences in the performance of signs by multiple users. Each DTW pass centers and sizes the sign using a different set of properties to generate multiple scores that can be combined to provide a better measure of similarity.

The two methods are evaluated using a vocabulary of 1,113 signs in both user-dependent and more realistic user-independent experiments with fluent signers. While

either method alone achieves an improvement in accuracy, particularly on subjects who perform the signs with large variation from the models, the combination of both techniques provides the best and most significant results. Finally, an improvement in accuracy is demonstrated on actual users of the dictionary system, who are unfamiliar with American Sign Language.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This thesis presents work towards a computer vision-based Large Vocabulary American Sign Language (ASL) video dictionary system. We propose novel methods for significantly increasing the accuracy of automated sign search systems by improving sign representation and similarity measure. This chapter explains the problem at hand and introduces the need for a better sign language dictionary system. A list of the contributions and a brief overview of the thesis is given.

1.1 The Need for Sign Language Dictionaries

Users of written languages have the distinct advantage of being able to quickly and easily search for the meaning of an unknown word; the sortable nature of alphabets affords them this opportunity. A simple Internet or printed dictionary search provides results in short time. Sign languages, however, lack this inherent sortability, as there is no obvious way to assign some type of order to a series of motions and hand shapes. With an estimated 500,000 to 2,000,000 users of American Sign Language in the United States alone [1, 2], many people are at a disadvantage. Students of a sign language would like to be able to look up the meaning of a sign they encounter with which they are unfamiliar. Users of sign languages, on the other hand, know what it means to them, but are unsure of the English translation. There are many English to sign dictionaries that are relatively easy to use, but the reverse is not true. While sign to English dictionaries do exist, the lack of easy sign sortability makes them cumbersome to use.

The American Sign Language Handshape Dictionary addresses the problem by categorizing its approximately 1,900 signs into 40 basic hand shapes [3], but this requires the user to first correctly identify the initial hand shape and then sort through a large number of signs. Assuming the signs are uniformly distributed into the various hand shape categories, the user may need to sort through nearly 50 illustrations showing hand shapes and positions with arrows indicating movement; it can be difficult to decipher what is occurring in these drawings. The Handspeak online ASL-to-English Dictionary [4] requires the user to first categorize the sign into one of 44 hand shapes; second, categorize the movement of the sign into one of 18 general movement categories; and, finally, categorize the location of the sign into one of 12 general locations. Still others require the addition of a hand orientation category.

ASL dictionaries are beginning to include video examples online or on a DVD, eliminating the need to decipher the illustrations, but it remains a time-intensive and non-trivial task to search for the meaning of an unknown sign. It is evident that there needs to be a better way to determine this meaning. This thesis presents work on a system that offers users and students of sign languages a more natural way of sign meaning search: by actually performing the sign.

1.2 Requirements for a Sign Language Dictionary

It is important to determine what capabilities a system that allows for visual sign search should have and to define a set of user requirements that make the system acceptable to the general public. Existing sign language dictionary systems exhibit both good characteristics and features that need improvement, and this thesis addresses many features that are lacking. For this dictionary system, we have identified the following required features:

1. The system must work visually by allowing the user to perform an unknown sign in front of a camera or other sensor – addressed throughout this thesis.
2. The system must achieve reasonable accuracy. It should maximize the number of results in the top 20 matches, since the user would likely not want to view more results than that. This is the primary focus and contribution of this thesis and is addressed in Chapters 3, 5, and 7.
3. The system must have automatic detection and tracking of the hands. No manual annotations to initialize a hand tracker should be required – addressed in Chapters 4–6.
4. The system must be fast, since users should not have to wait for results. This requirement is addressed in Chapter 6 and is a vast improvement over the previous variant.
5. The system must be easy to use and as intuitive as possible – addressed in Chapter 6.
6. Rather than return a single most similar sign, the system must return a list of signs ranked according to similarity and allow the user to browse the results.
7. The system must present video examples of the matched signs for user verification – addresed in Chapter 6.
8. The system must require as little user intervention as possible – addresed in Chapter 6.
9. The system must have large vocabulary – addressed by using a vocabulary of 1,113 signs as described in all recognition experiments.

1.3 Problem with Large Vocabularies and Limited Training Sets

It is a trivial task to design a gesture set with a relatively small number of sufficiently dissimilar gestures so as to make recognition easy. But the goal of sign

language recognition is not to make a gesture set that works well with a given method, but to design a method that works well with a pre-existing, well-established vocabulary. The latter is a far more difficult task.

A gesture set, like a sign language, with a sufficiently large vocabulary is bound to exhibit a great degree of similarity in hand trajectories between many gestures. Many one-handed signs are static, meaning the hand does not really move much. Instead, the difference lies in the hand shape. Often, this difference can be as little as the thumb being extended vs. not being extended, as is the case with the signs for the numbers 14 and 15. Figure 1.1 shows examples of four signs that occur in roughly the same region and have very little movement. These signs generally have an arbitrary hand trajectory caused by unintentional movement of the hand, resulting in wide variation in the trajectory that may confuse trajectory-based recognition methods. Furthermore, in user independent systems, there will be additional large variability in the performance of gestures, both in size and position, due to the personal styles of individual signers. This can result in unintended similarity between gestures.

The similarity in gestures is difficult to deal with. This thesis does not incorporate any hand shape information into the recognition algorithm and, instead, focuses on improving the trajectory matching process by somewhat relaxing the assumptions of where a sign occurs in the signing space and how large the signs may be. It would be relatively simple to incorporate some hand shape descriptors into the process, given a reliable method to segment the hands.

When one has few training data, it is difficult to create an algorithm that generalizes well beyond the training set, and they tend to overfit the training data, especially when only a few signers are performing the set. Machine learning and statistical methods certainly prove their worth in gesture recognition as the literature shows, as long as sufficient training data are provided to generate the required

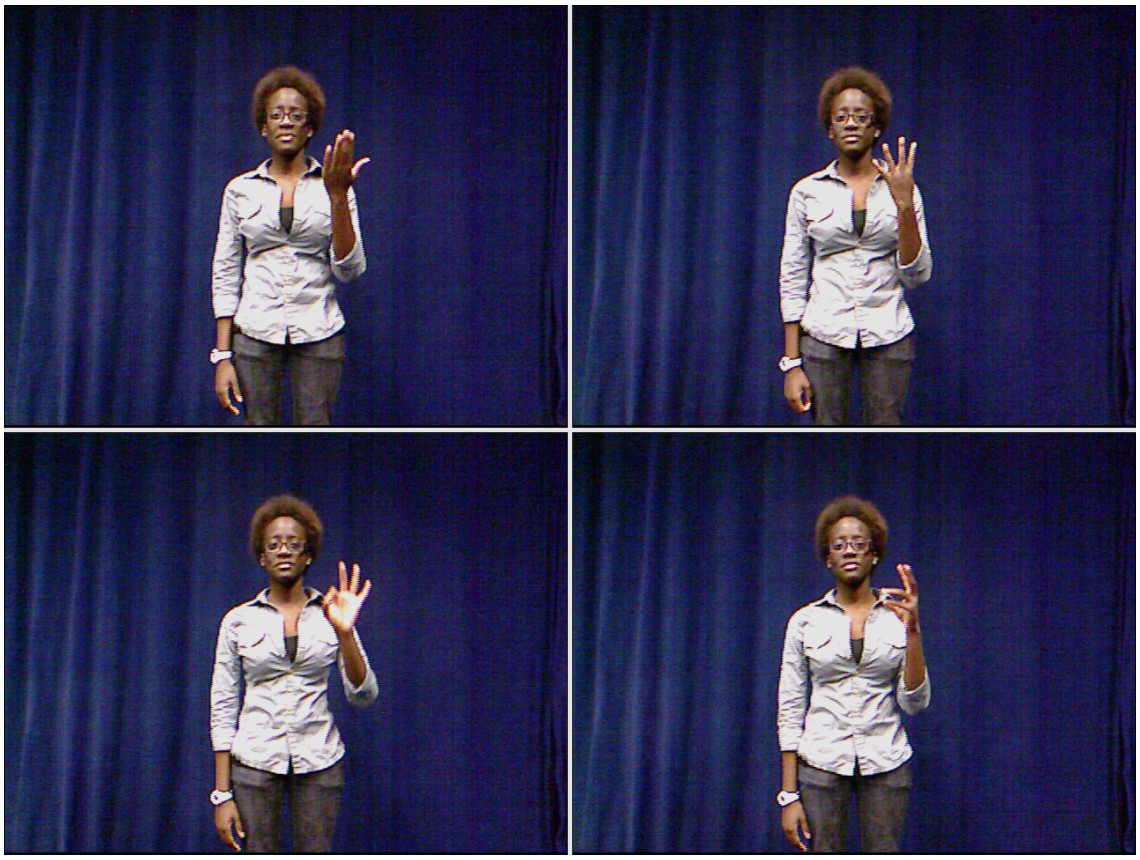


Figure 1.1: Examples of sign similarity. The position is roughly the same, but the hand shape differs.

probability distributions or training and validation sets that can ensure generalization to other individuals not in the training set. Generating and annotating these data, especially for a large vocabulary, however, is expensive in both fiscal and manpower terms. It is for this reason, that this thesis introduces novel methods for increasing accuracy in large vocabulary systems with minimal training data by improving similarity measures in exemplar-based recognition.

1.4 Main Contributions

Given the necessarily visual nature of sign languages, it makes sense to approach the problem from a computer vision perspective. The concept behind the dictionary

system is that the user performs an unknown sign in front of a camera or other sensor, the system extracts features from the video frames that represent the sign and compares them to the features of the signs in the database to rank them according to similarity. The user is then presented the ranked list of signs and their meaning with a video example of each that he or she can view to find the desired sign and its English equivalent. Though the focus of this thesis is on a large vocabulary American Sign Language (ASL) video dictionary system that meets the requirements outlined in section 1.2, the techniques are applicable to any gesture set of arbitrary size. Particular emphasis will be given to increasing the accuracy of such systems, where there is large variability in the performance of signs among users.

1.4.1 RGB-D ASL Dataset

Since there is a lack of publicly available large vocabulary gesture datasets that have full annotations, this thesis contributes an RGB-D body part detection and ASL recognition dataset along with benchmarks for the evaluation of hand detection and sign recognition methods. This growing dataset currently consists of 1,113 unique signs performed by multiple fluent signers, and is recorded with a Microsoft Kinect. Full annotations are provided, including information about the signs themselves, such as lexicon, or meaning, sign handedness—one-handed or two-handed—and start and end frames. Also provided are manual annotations in the form of bounding boxes for the head in the first frame of each sign, as well as those for the hands in every frame of each sign, points indicating shoulder and elbow locations in each frame of the signs, and indications of occlusion of the joints. Kinect skeleton joint positions are also provided in both 2D pixel coordinates and 3D real world coordinates. Though currently incomplete, the dataset continues to expand with additional annotations, as

well as signers. The final vocabulary will consist of some 3,000 ASL signs performed by multiple signers.

1.4.2 Novel Methods to Improve Similarity Measures

The second focus, where the novel contributions are made, is on improved sign representation and match accuracy. Any gesture and sign recognition system needs an effective similarity measure for sign comparison. This work explores modification of the exemplar-based recognition method that is currently used in the system to include novel features and a way to combine multiple similarity scores. The work presented in the following chapters will make the following two contributions with respect to this focus:

1.4.2.1 Intra-Class Variation Modeling (ICVM)

Different signers will lend their own personality to a sign, resulting in wide variation in some geometric and positional properties of the hand trajectories. Some signers exaggerate the gestures, while others are more subdued in their performance. Some perform the sign in a large space, others in a small space. Size normalization and choice of a coordinate system can account for these differences to a certain degree, but this variation still remains a problem. ICVM models these differences across same signs and allows for researchers to generate an indication of likelihood that two signs are a match given the differences between them in these properties. These likelihoods can then be used to filter potential matches using, for example random forests, SVMs, or a cascade weak classifiers, or, as is done in this work, as part of the similarity measure itself. By weighting and combining these likelihood features, we can substantially improve recognition accuracy.

1.4.2.2 Multiple-Pass Dynamic Time Warping (MP-DTW)

MP-DTW is the second novel method presented in this work to handle variations in sign performance across users. The former variant of the dictionary system expresses the positions of the hands in a single coordinate system with the center of the face at the origin and sizes the trajectory based on the size of the face. While this does a relatively good job at making recognition somewhat scale and video frame position invariant, it does not do a good job at accounting for differences in the positional and size variations within the signing space. Each user will perform a sign in a somewhat different place relative to the face and at a different size. MP-DTW helps account for these differences by re-centering and resizing a sign using different origins and size normalization features than the face. For example, one pass may center the sign on the center of the right hand trajectory and size the sign based on the width of the trajectory, while another centers on the left hand trajectory and sizes the sign using the trajectory height. This generates multiple classifier scores for the match, each focusing on a different aspect of the sign, that can be combined into a single similarity measure.

1.5 Thesis Overview

Chapter 2 examines the existing work in several human activity analysis areas—general activity and action recognition, generalized gesture recognition, and sign language-specific recognition—and their shortcomings. Chapter 3 presents the required background information for the understanding of the similarity measurement and gesture representation of this sign recognition system and the concepts behind the proposed methods.

Chapter 4 introduces the new RGB-D dataset of ASL signs and their annotations and provides an analysis of the Kinect's hand tracking capabilities. The dataset and hand tracking benchmark can be used for body part detection research as well as gesture recognition projects. The dataset is continually evolving and will ultimately consist of multiple examples of some 3,000 signs performed by several subjects with varying signing styles.

In Chapter 5, we establish a baseline recognition method and benchmark accuracy on a test set from the dataset described in Chapter 4. It gives us a recognition system foundation and a way to evaluate the methods presented in Chapter 7.

Chapter 6 presents the dictionary system along with a set of actual usage experiments, illustrating the benefits of the system with users unfamiliar with ASL and the system itself. The chapter also highlights the benefits and superior performance over the last variant of the ASL dictionary system, both in terms of accuracy and speed.

Chapter 7 presents the two novel methods for increasing recognition accuracy based on variations in geometric properties of signs inherent in user-independent systems. The first, Intra-Class Variation Modeling, provides the set of features that serve as a likelihood that a test gesture belongs to the same class as a model gesture given their differences in these properties. The second proposed method leverages the fact that different users will perform signs in varying positions in the signing space with differing amount of sign exaggeration to improve recognition rates. These two methods are designed to work in systems that contain a large vocabulary but a limited training set, since our training set currently consists of only three examples per unique sign.

In Chapter 8, we demonstrate the potential of the two methods in a series of user-dependent and user-independent experiments and achieve significant improve-

ments in accuracy using both manual and RGB-D annotations for hand and head locations. We also show that the methods out perform competitors in large vocabulary systems with few training examples. The chapter additionally presents experiments that apply the techniques to the signs recorded by actual users of the system discussed in Chapter 6 and demonstrate their benefit in real-world scenarios. Finally, we present discussion of the statistical significance of the accuracy improvements, as well as the effect of the number of variation modeling features used.

CHAPTER 2

RELATED WORK

2.1 Action and Activity Recognition

Most recent work has been in action and activity recognition, some from static images [5, 6], others from video [7, 8]. Most RGB methods are rooted in parts-based models, consisting of a collection of parts, for example forearm, upper leg, etc., and a model of their configuration. Tian et al. [7] extend Felzenszwalb's deformable parts model [5] into a temporal dimension to use the concept on video. Hierarchical approaches have also proven successful. Wang et al. take such an approach to the parts model, in which any part can consist of a group of subparts [6]. Similarly, Ma et al. employ a two level system and track parts at the whole body level and subpart level through time, calling them space-time segments [8]. Song et al. approach the problem instead through a temporal hierarchy and extract features at different temporal resolutions [9].

Others leverage the depth-sensing capabilities of RGB-D to recognize actions. Some work at the pixel level. Lu et al. developed a depth feature based on a set of comparisons between the depths at pairs of pixels at different points in time and then use an SVM classification system [10]. Others instead rely on the Kinect skeleton detector output of joint positions or the angles between joints. Vemulapalli et al. model relative geometry between body parts that are not necessarily directly connected [11].

Though it is impressive what these action recognition works achieve, they tend to focus on classifying small vocabularies of general actions, for example recognizing golf vs. gymnastics, rather than one gymnastics move vs. another. This is somewhat

analogous to identifying a video as sign language vs. juggling or aircraft marshaling, rather than identifying specific signs. Some action recognition works do test their methods on gesture datasets [12, 13], but the vocabularies are limited, and the methods are generally not directly applicable to our large vocabulary gesture sets.

2.2 Generalized Gesture Recognition

Other research focuses on general gesture recognition and is more applicable to this work. The gesture sets may be created specifically for this task and can be chosen so as to minimize similarity between classes. Long Short-Term Memory (LSTM) Recurrect Neural Networks (RNNs), one of our comparison methods in Chapter 8, have seen success, and Google has incorporated one into Android to recognize keyboard gestures [14]. Hidden Markov models, also one of our comparison methods remain popular, as can be seen throughout the literature and this thesis Section.

There is an abundance of computer vision gesture recognition research employing RGB cameras. Many of these methods are model-based, using Hidden Markov Models [15, 16, 17], or alternative approaches such as recognizing motion patterns from hand trajectories using Time Delay Neural Networks [18] and classifying hand shapes using a recursive partition tree approximator [19]. All of these methods use a small vocabulary of signs (less than 100 signs) and have unknown potential for scalability.

There has also been work in RGB-D generalized gesture recognition. In early Kinect research, Doliotis et al. reach 95% gesture recognition accuracy in cluttered scenes but only employ a simple vocabulary of 10 digits drawn in space with the hand and make the assumption that the hand will be the closest body part to the sensor [20]. This is often not the case with ASL.

With the recent ChaLearn Gesture Challenge [21], there have been a number of works in one-shot learning, in which a single training example is used per class. Wan et al. propose utilizing the multimodal data of the Kinect to create a 3D Enhanced Motion Scale-Invariant Feature Transform (3D EMoSIFT) to describe both motion and appearance for use in a Bag of Features style approach [22]. Konečný et al. propose using Dynamic Time Warping and a combination of Histogram of Oriented Gradients (HOG) features to describe appearance of the depth images and Histogram of Optical Flow (HOF) features to describe the motion [23]. Jiang et al. propose a 3 classifier hierarchical approach that progressively eliminates candidate matches with each layer [24]. Fanello et al. use a sparse representation approach of appearance and motion from RGB-D video using HOG and HOF with an SVM classification system for real time recognition [25]. Pfister et al. combine a strongly supervised model (one-shot learning) and multiple weakly supervised examples, using a Global Alignment Kernel [26] for sequence alignment [27]. Goussies et al. employ decision forests and transfer knowledge from training on one gesture set to other similar gesture sets. Pitsikalis et al. use motion, handshape, and audio streams from the Kinect to generate multiple scores and hypotheses that are weighted and combined into a single best hypothesis [28].

The experiments for the ChaLearn gesture challenge, however, were user dependent. The multiple performances of the gestures occur in the same scene and position in the video frame, and many of the methods take advantage of this fact and use global level features on entire video frames or large regions of interest in video frames. Others assume that the gesture performer provides the only motion found in the video frames. These assumptions are unrealistic in our work, and it is unclear how well these methods generalize to handle inter-user variation and scale from the 8 to 12 gesture ChaLearn vocabularies to ours of 1,113.

2.3 Sign Language Recognition

2.3.1 Fingerspelling, Sign Spotting, and Continuous

A third focus is on developing methods for well-established gesture sets, such as sign languages. One branch of work is in continuous sign language recognition and fingerspelling, or the spelling out of words with a signed alphabet. Kim et al. propose a method to break a video into variable length segments, using letter transition probabilities, hand shape similarities scores, and a semi-Markov Conditional Random Field (CRF) to identify the string of letters in fingerspelling videos [29]. Some research focuses on distinguishing between signs and movement epenthesis, or transitions between signs, to improve recognition [30, 31, 32, 33]. Nayak et al. focus on unsupervised learning of signs by looking for patterns in multiple videos of continuous sign language that can be automatically extracted as signs or subunits of signs [34, 35, 36]. Kelly et al. also work toward automatic learning and extraction of 30 signs in continuous sign language videos with accompanying text translations [37]. To learn the signs, they use Multiple Instance Learning (MIL) on sets of videos that contain the desired sign more often than any of the other signs; their method does not require the usual set of videos that do not contain the desired word. The method, however, does require the user to wear colored gloves so that they can accurately segment the hands.

Continuous sign language recognition and sign spotting methods can require large amounts of training data, generally use small vocabularies and are not particularly useful with our dictionary system goals, though experimentation with sign spotting could be warranted in an attempt to eliminate the user-provided temporal segmentation of the sign as is currently required.

2.3.2 Segmented Sign Recognition

Other research focuses instead on classification of individually segmented signs. Gloves were once a commonly used input source for sign language recognition [38, 39, 40]. Yao et al. [38] and Liang et. al [39] collected more than 1,000 signs and 250 signs by gloves, respectively. Both systems used HMMs to model the signs. Yao et al.[38] proposed a pre-processing method, called One-Pass pre-search, to speed up the recognition process. Wang et al. [40] studied how to track the movement of fingers through video of a glove with differently colored fingers and areas. Sandjaja et al. achieve 85.52% accuracy in a Filipino Sign Language number recognition system but require the user to wear a multi-colored glove [41] to automate hand and finger location and tracking. It is undesirable for the system described in this thesis to require the user to wear any special equipment for sign recognition; it is to be as user friendly as possible.

One popular intuitive method is to segment a sign into motion or other types of subunits and then use HMMs to model the temporal changes in subunits throughout each sign. Cooper et al. provide a comparison of two subunit methods, experimenting with both an HMM and Sequential Pattern Boosting [42]. Using an HMM only on frames their method designates as high-ranked key frames, Wang et al. achieve good results in user-independent tests on their large Kinect dataset of Chinese Sign Language (DEVISIGN) using a vocabulary of 1,000 signs [43, 44]. Their work is perhaps most similar to what we are trying to do with out ASL recognition research. HMMs work well with enough training examples to learn the transition probabilities, but our experiments show that 3 examples per class, as found in our dataset, is insufficient.

Instead of taking an HMM approach, Han et al. break the signs into subunits based on hand motion discontinuity and generate a codebook of medoid segments for

each sign [45, 46, 47]. They train classifiers for the subunits instead of the signs to use on the video input, since there are far fewer types of subunits than there are signs. Once the codebooks are generated, they use AdaBoost to train a classifier for each signs. While they claim scalability, they only tested with a vocabulary of 20 signs, and it is unclear whether the experiments were user independent.

Lichtenauer et al. propose separating the alignment and classification portions of recognition by using Statistical DTW (SDTW) for alignment and then a separate algorithm for classification [48]. Rather than identifying the sign as belonging to the class with the highest likelihood according to SDTW, they hypothesize that some of the transition probabilities that played a role in alignment (i.e. those related to going from rest into the sign movement) should be ignored, and thus use a separate classification scheme on the aligned features. They experimented with a relatively small vocabulary of 120 signs with 75 examples of each.

Much of the research involves vocabularies of limited size or requires user-dependent tests to achieve high accuracy. Zieren et al. achieve 99.3% accuracy in user-dependent sign language recognition experiments using a 232 sign vocabulary; the accuracy, however, decreases to 44.1% in user-independent experiments with a vocabulary of 221 signs [49]. Similarly, Kadir et al. achieve high accuracy with a vocabulary of 164 signs, but also use the same signer for the training and testing sets [50]. User dependence is not a realistic requirement in a sign video dictionary system, since it is an unacceptable expectation that the user pre-perform the signs in the vocabulary and then manually annotate joint positions.

In the work on which this thesis builds [51, 52, 53, 54], the authors use vocabularies of comparable size and ensure user-independence, but require the user to provide hand locations either for each frame or for the first frame to initialize a hand

tracker. The work presented herein automates the process of hand detection and relieves the user of this responsibility.

There is also a body of research using the Kinect or similar RGB-D cameras for gesture and sign language recognition, but these studies also tend to use limited vocabulary size and gesture complexity. Agarwal and Thakur achieve good results using a static hand gesture vocabulary, consisting of Chinese Sign Language signs for digits [55].

Zafrulla et al. conduct a Kinect-based ASL recognition feasibility study in which they recognize 60 distinct, simple phrases of 3 to 5 signs using a 19 word vocabulary [56]. The authors conducted both seated and standing tests. They achieve word and sentence recognition accuracy of 74.48% and 36.2%, respectively, for seated tests and 73.62% and 36.3% for standing tests.

Pedersoli et al. explore real-time gesture recognition using a vocabulary of 16 relatively simple one-handed gestures and achieve better than 70% accuracy [57]. However, it requires an open palm, forward-facing orientation for hand segmentation and assumes the hand is the closest object to the camera for hand pixel clustering to work. The vocabularies for these works are clearly too small for a sign language dictionary system.

The methods and experiments presented in this thesis were designed with the shortcomings of this previous work in mind and more accurately simulate a real-world usage scenario, thereby providing a more realistic baseline measurement. The experiments use a large vocabulary of 1,113 signs, ensure user-independence, and require no special gloves or markers to track the hands; it is as automated as possible.

CHAPTER 3

BACKGROUND

3.1 Dynamic Time Warping Review

Dynamic Time Warping (DTW) is a time series analysis technique, easily implemented with dynamic programming, that creates an optimal alignment between to sequences [58]. In our case, the sequences are the representations of two signs—a model sign, M, and a query sign, Q. DTW creates a minimal cost warping path between two signs by effectively matching frame by frame what is occurring in the test and model videos, as described by the feature vectors introduced in Section 3.2.2. The score for matching model sign M to query sign Q given any warping path $W = \big((q_1, m_1), ..., (q_{|W|}, m_{|W|})\big)$ of length $|W|$ is the summation of the individual costs $c(Q_{q_i}, M_{m_i})$ to match query frames q_i to model frames m_i in the warping path:

$$C(W, Q, M) = \sum_{i=1}^{|W|} c(Q_{q_i}, M_{m_i}). \tag{3.1}$$

The base DTW score D_b between query Q and example M is provided by the lowest cost of all warping paths:

$$D_b(Q, M) = \min_{W} C(W, Q, M). \tag{3.2}$$

Section 7.4 on Multiple-Pass DTW that focus on various aspects of the hand trajectories builds on this concept and generates multiple scores to be linearly combined with this base score. DTW enforces three constraints to determine the optimal warping path:

1. **Boundary constraints**: $q_1 = 1$, $m_1 = 1$, $q_{|Q|}$, $m_{|W|} = |M|$. This ensures, in our case, that the first frame of the query matches the first frame of the model.

2. **Monotonicity**: $q_{i+1} - q_i \geq 0$, $m_{i+1} - m_i \geq 0$. The frame matches can only progress forward or remain still; there is no backwards movement in time.
3. **Continuity**: $q_{i+1} - q_i \leq 1$, $m_{i+1} - m_i \leq 1$. No frames numbers in either the query or model sign will be skipped.

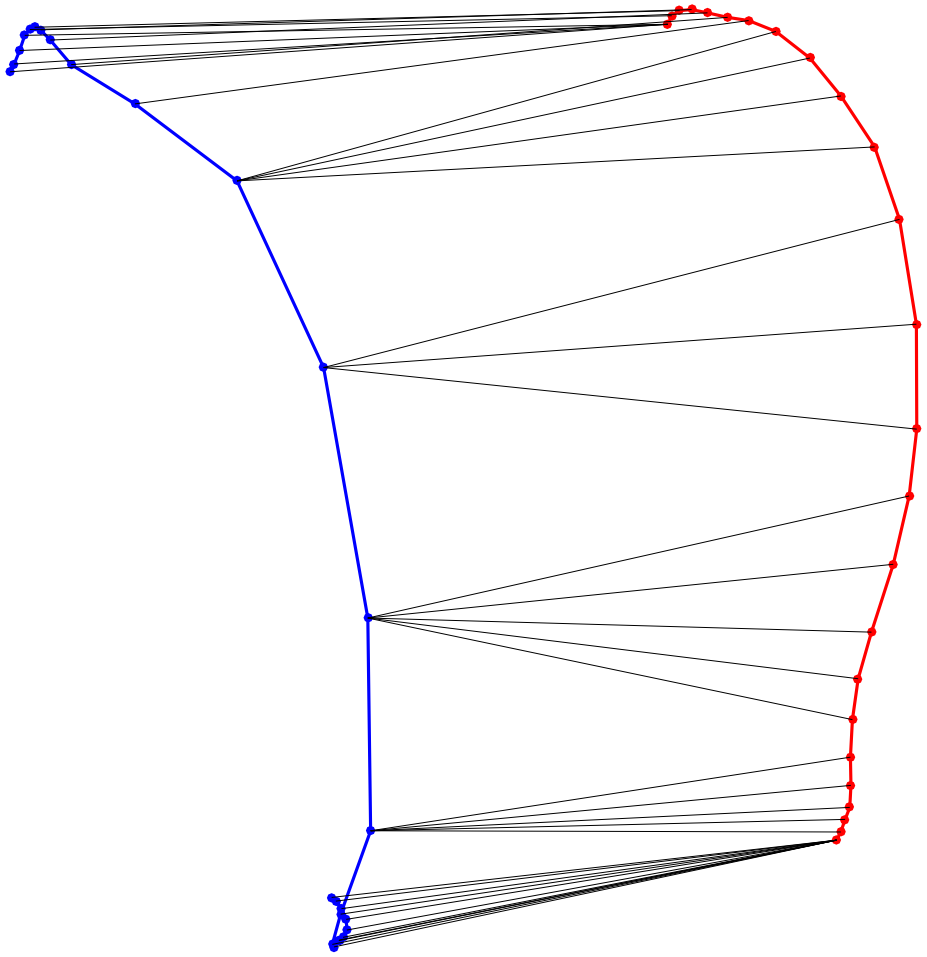


Figure 3.1: Example DTW alignment between two same-class sign trajectories using only hand positions.

Figure 3.1 shows an example alignment, using hand positions only, between the dominant hand trajectories of two examples of the ASL sign for *cheap*. The solid circles represent the positions of the hands throughout the sign. The lines indicate

frame matches. It can be seen that several hand positions on the left side match with many hand positions on the right side. This indicates that one signer performed the gesture at a faster rate. The sum of the squared Euclidean distances between these frame matches becomes the DTW score.

3.2 Sign Representation

We need an efficient way to describe a sign, since it would be computationally inefficient and, due to differences in the size and location of signs in a video, not particularly productive to vectorize the pixels of each frame and run it through the DTW algorithm. We need a set of descriptors that represent what is occurring at each time t in a sign, both in terms of positions of the hands and of motions. For this task, we use the feature vector described in [52], slightly modified to account for the differences in RGB and RGB-D technology. Since automation is a key requirement of this sign language video dictionary system, as detailed in Chapter 1, the feature extraction and vector construction method is modified to use Kinect-provided joint positions, rather than those derived from the manually annotated hand and face bounding boxes.

3.2.1 Face-Centric Coordinate System

Since the recognition system does not have control over where the user performs the sign in the video frame, it is important to choose a coordinate system to describe hand positions that allows for translation invariance. There are several options to choose from to be at the origin of the coordinate system. Past work has shown the center of the face to serve well. Therefore, before constructing any feature vectors to describe each frame of the sign, the hands—and other body parts, if being used—need to expressed in this new coordinate system by subtracting the position of the

face center. This effectively centers the sign viewpoint on the user's face and ensures translation invariance.

3.2.2 Extraction of Features

To represent a sign, a feature vector based on 2D hand position information is built for each video frame that describes what is occurring at that point in time. For the DTW pass, the hand positions are expressed in the face-centric coordinate system for the reasons outlined above. For one-handed signs, the position of the non-dominant hand is set to $(0, 0)$ so as not to contribute to the DTW score. The following features compose the vectors for each frame t of sign video X:

1. $L_d(X, t)$: Pixel position of the dominant hand.
2. $L_{nd}(X, t)$: Pixel position of the non-dominant hand.
3. $L_\delta(X, t) = L_d(X, t) - L_{nd}(X, t)$: Position of the dominant hand relative to the non-dominant hand.
4. $O_d(X, t)$: Motion direction, expressed as unit a vector, from frame $t-1$ to frame $t + 1$ for the dominant hand.
5. $O_{nd}(X, t)$: Motion direction, expressed as unit a vector, from frame $t - 1$ to frame $t + 1$ for the non-dominant hand.
6. $O_\delta(X, t)$: Direction of change for L_δ from frame $t - 1$ to frame $t + 1$, expressed as a unit vector.

There are four main types of two-handed signs: signs in which the hands move in a symmetric manner, signs in which the hands move in an anti-symmetric manner, signs in which the hand movement is a combination of the other types, and signs in which one hand is static and the other moves. In the above feature descriptions, the dominant hand is the hand that will be moving.

The feature vectors for each frame are then combined into a single matrix to describe the sign, so that each row represents a single frame of video. Given the features described above, the DTW local cost of matching Query frame number q_i to model frame number m_i, as described in Section 3.1, now becomes a weighted linear combination of the squared Euclidean distances between the six components of the features:

$$\begin{aligned} c(Q_{q_i}, M_{m_i}) = & s_1 \|L_d(Q, q_i) - L_d(M, m_i)\|^2 + \\ & s_2 \|L_{nd}(Q, q_i) - L_{nd}(M, m_i)\|^2 + \\ & s_3 \|L_\delta(Q, q_i) - L_\delta(M, m_i)\|^2 + \\ & s_4 \|O_d(Q, q_i) - O_d(M, m_i)\|^2 + \\ & s_5 \|O_{nd}(Q, q_i) - O_{nd}(M, m_i)\|^2 + \\ & s_6 \|O_\delta(Q, q_i) - O_\delta(M, m_i)\|^2, \end{aligned}$$

where weights $\{s_1, ..., s_6\}$ are empirically determined on a validation set.

Figure 3.2 shows the alignment of the same trajectories as Figure 3.1 but using the full feature vector described above. By including motion direction and the other components, the alignment better matches similar areas of the trajectories. Rather than matching based purely on Euclidean distance between points on the trajectory, it allows matching the curvature of the trajectories as well.

3.2.3 Sign Size Normalization

Just as the system does not have control over the position in the video frame where the user performs the sign, it also does not have control over the signer's distance to the camera or the resolution of the camera. When the user is further away from the camera, the same trajectory will appear smaller, so it is important to normalize the size of the sign to make the method scale invariant. For the experiments

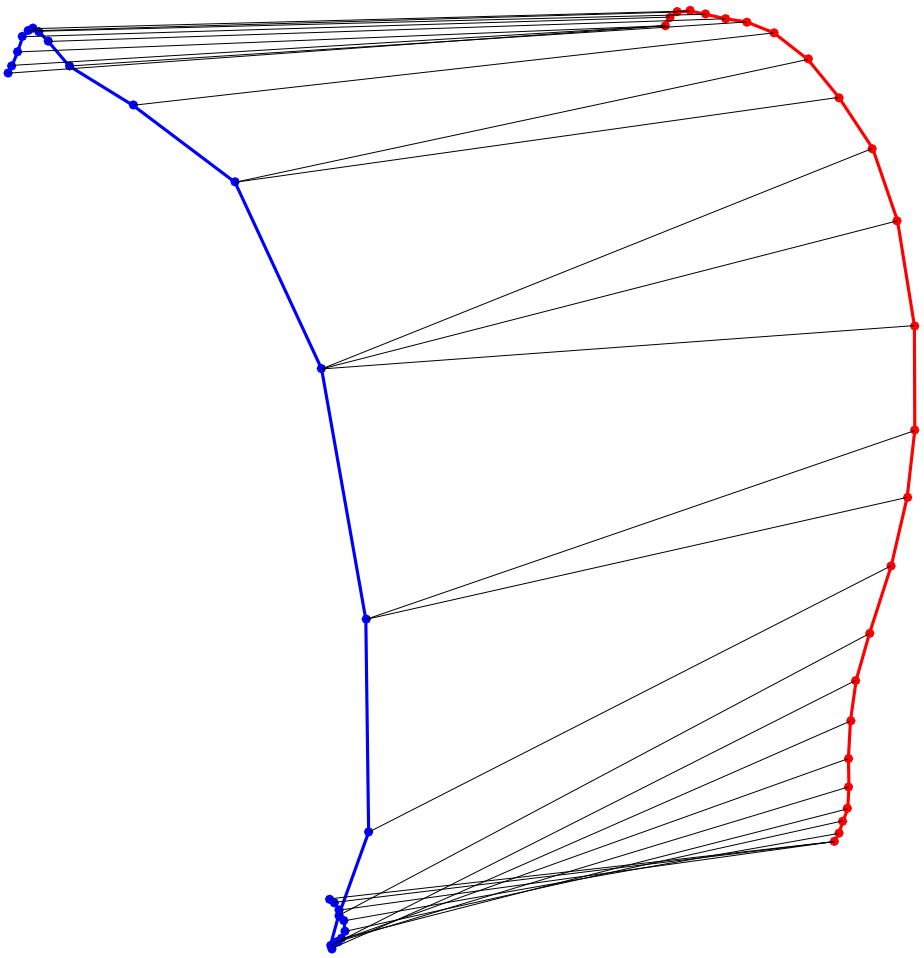


Figure 3.2: Example DTW alignment between two same-class sign trajectories using the full feature vectors.

using the manual annotations of the hand positions, the sign is size-normalized so that the diagonal of the face bounding box is 1. This technique has proven to provide a good representation of the coordinate space.

Other experiments use the positions of the hands and head provided by RGB-D skeleton detectors, so we do not have access to a bounding box for the face. We can use a face detector, but experimentation has shown that the bounding box size varies greatly, even without the person adjusting their distance to the camera. We instead want to use information about joints that are easily located by the skeleton detectors.

Resizing the sign based on the head-neck distance is one such option and was used in the early experimentation with the system. However, it makes sense that shoulder width would provide a better indication of the size at which a user would perform a given sign. A person with broad shoulders is more likely to perform the sign in a larger space, due to longer arms, while their head-neck distance is likely nearly the same as a narrow-shouldered individual.

The Kinect does a good job at locating the shoulders, so they they are what the system currently uses. Due to the slight joint location stability problems inherent in the skeleton detection algorithm, the sign is size-normalized so that the average non-zero shoulder width throughout the sign is 1.25. This provides a more consistent resizing distance by reducing the effect of outlier shoulder positions, and ensures that frames in which the shoulders are not detected (and set by default to 0,0 in our annotations) are not included in the resizing calculations.

3.2.4 Frame Length Normalization

One issue with using DTW for gesture recognition tasks is that it is biased toward shorter model signs. With fewer frames to match, fewer error distances are added to the sum of local frame distances in Equation 3.1. In order to eliminate this bias, we can make all model signs the same number of frames using interpolation. Experimentation has shown that bicubic interpolation on the feature matrix produces the best results. A frame normalization length of 25 has also been empirically determined to provide the best combination of accuracy and time required by the dynamic programming DTW algorithm.

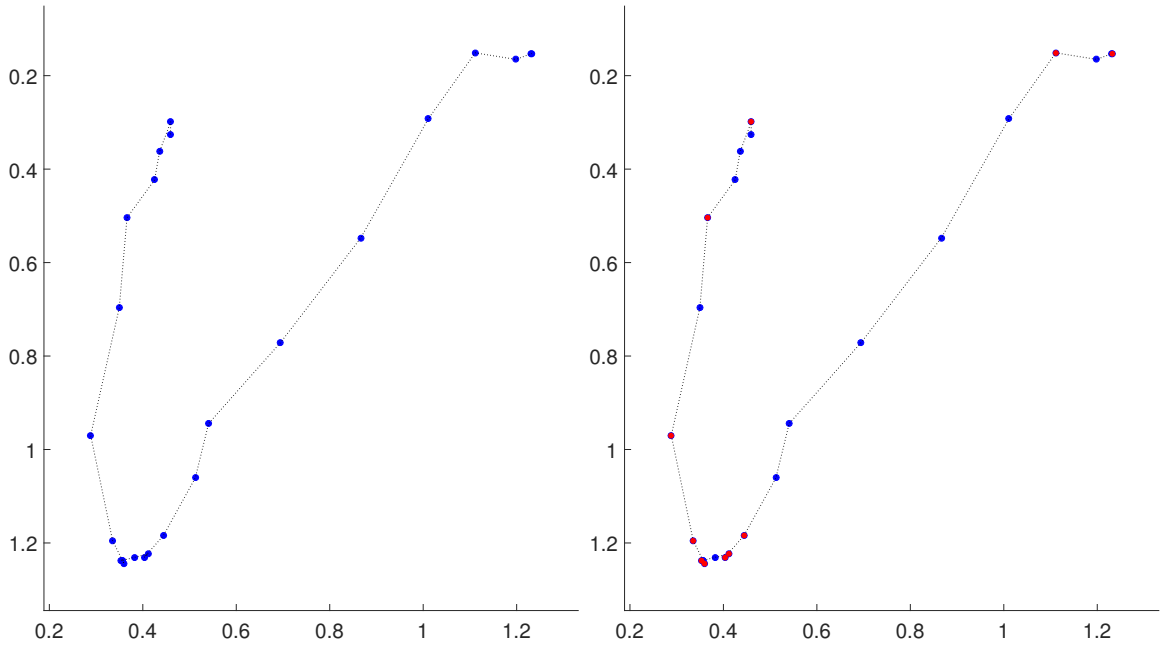
3.3 Convex Hulls

This section briefly introduces the concept of convex hulls and the motivation for using them in this work. Given a set of points in space—2-Dimensional space in this work—if we were to stretch a rubber band around the points and release it, the rubber band would contact certain points and form a polygon that encompasses the entire set of points with a minimal set of vertices. The points that are in contact with the rubber band become the vertices of the polygon; all other points are interior points and do not define the polygon.

Figure 3.3 illustrates this concept with the ASL sign for the English word *check*. If we remove the temporal aspect of the sign, we can look at the trajectory as a set of points (i.e. hand positions) in space. In 3.3a, the entire set of these positions from the entire trajectory are shown for the dominant hand. In this thesis, we are interested in various properties of the general shape of the closed trajectory. If we stretch the rubber band around the points, certain points will define the convex hull polygon, shown as red points in 3.3b and shown isolated in 3.3c. Figure 3.3d shows the resulting polygon with the extraneous interior points removed. It can be seen that the centroid of the convex hull better describes the center of the trajectory than does the center of the trajectory bounding box.

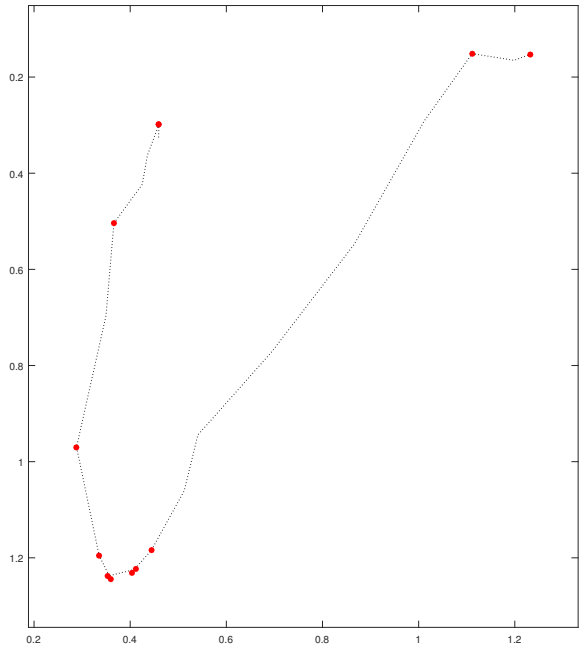
To understand the motivation for using convex hulls, consider figure 3.4. The trajectory forming the sign is shown as a blue line, while the hand positions at each frame are shown as blue circles. The red circle is the center of the trajectory bounding box, while the black circle is the centroid of the convex hull. The centroid provides a better indication of the true center of the trajectory shape than does the bounding box center. The method decribed in Section 7.4 explores the potential of centering the coordinate system on a position other than the face, as is done in the base DTW method. If we were to center the sign, for example, on the centroid or center of

the trajectory, we would potentially achieve a better DTW alignment. This figure demonstrates that the centroid would perhaps be a better choice for alignment, as it is closer to the true center of the trajectory.

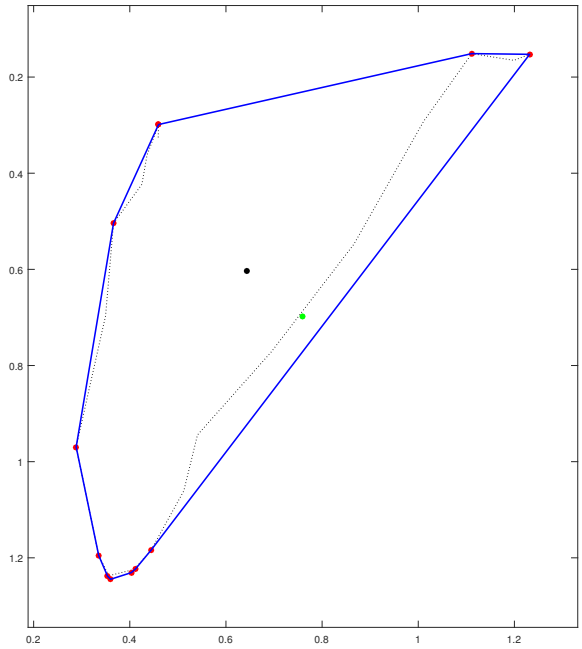


(a) Trajectory Points

(b) Convex Hull Points in Red



(c) Isolated Convex Hull Points



(d) Trajectory Convex Hull

Figure 3.3: Shows the development of the trajectory convex hull. The center of the trajectory bounding box is shown as a green circle, while the centroid of the convex hull polygon is shown as a black circle.

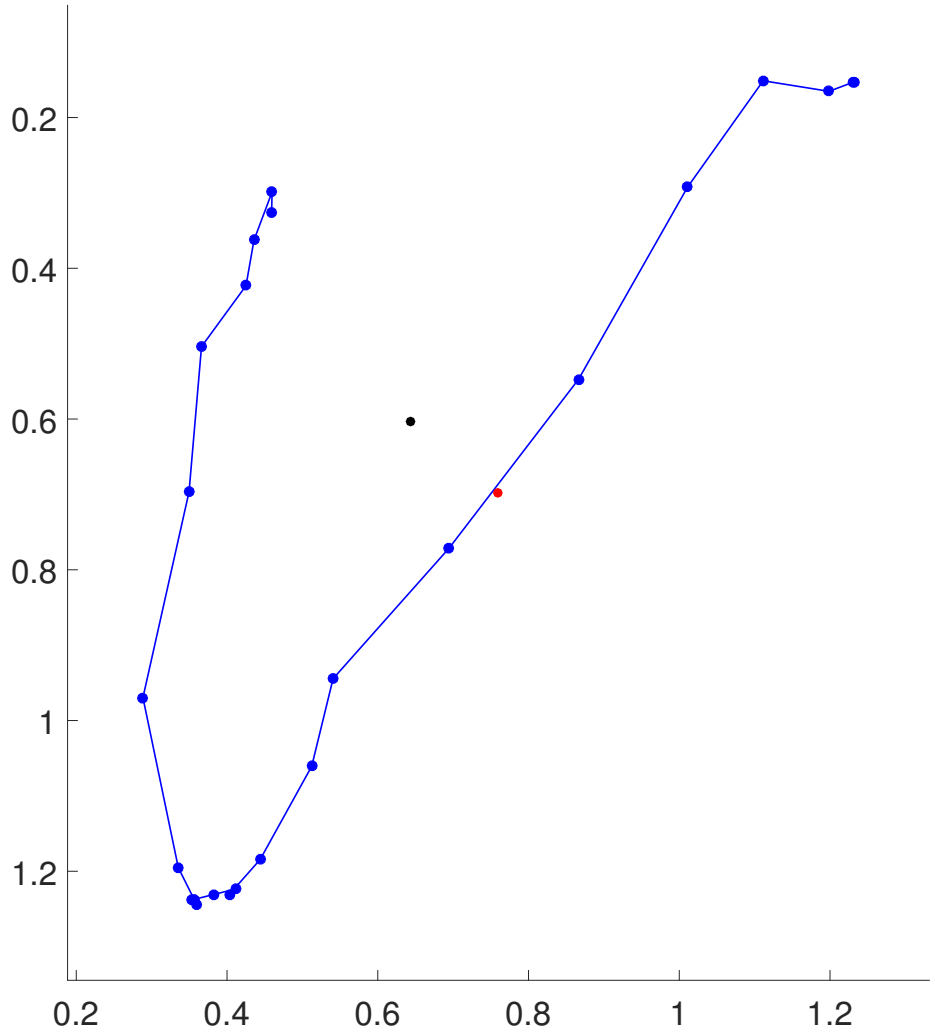


Figure 3.4: Plot of the center (red circle) of the trajectory bounding box and the centroid (black asterisk) of the convex hull centroid. The centroid is a better indication of the center of the trajectory.

CHAPTER 4

3D BODY PART DETECTION AND ASL RECOGNITION DATASET

4.1 Introduction

This chapter introduces a new 3D body part detection and ASL recognition dataset that will become part of the ASL video dictionary system and will be used to develop the new methods proposed in this thesis. A sign language video dictionary system like that described in [53, 52] or the updated version in this work necessitates a certain level of visual human-computer interaction. More specifically, it requires a vision system that is able to reliably detect and track a user's hands (and possibly other body parts), so that information about them—for example position, appearance, and movement—can be used to look up the meaning of the unknown sign. With the advent of the Microsoft Kinect in 2010, computer vision researchers were presented with an opportunity to utilize scene depth information, a capability previously only available with more expensive or cumbersome systems, such as laser depth sensors, stereo cameras, or multi-camera systems. The Kinect and its kin are thus usable in products that are more approachable by the average consumer and offer the potential to more reliably detect and track the hands using this scene depth information.

In addition to improving hand detection rates, incorporating information about the third dimension into gesture recognition tasks affords us a more accurate representation of what is actually occurring in the scene. A gesture is not merely a 2D, planar event. It has a 3D trajectory and thus, for the utmost accuracy in its representation, requires the third dimension information for trajectory matching. There is

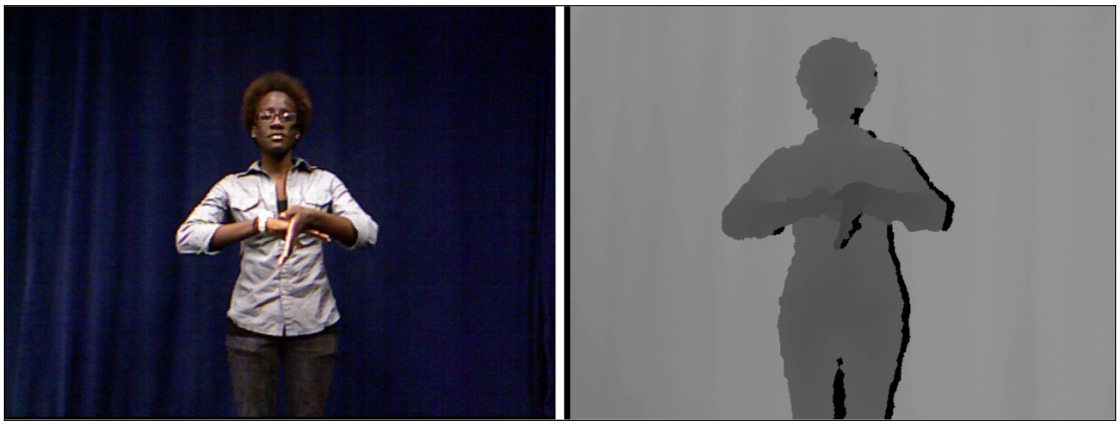


Figure 4.1: Sample dataset sign frame. Left: color video frame; Right: scene depth information.

a lack, however, of publicly available 3D ASL datasets. It is with these motivations that the dataset described herein is being created.

Reliable detection of the hands can be difficult in real-world sign recognition scenarios, and the dataset presented in this chapter allows researchers to develop new hand detection and tracking algorithms and experiment with both 2D and 3D gesture and sign recognition methods. A hand location accuracy benchmark is presented in this chapter that provides a baseline measurement to which researchers can compare their own hand detection and tracking methods.

4.2 Dataset-Specific Related Work

One of the highest quality video datasets useful for hand detection and gesture recognition research is the American Sign Language Lexicon Video Dataset (ASLLVD) [59]. It consists of a large set of recordings from multiple camera angles of the signs contained in the Gallaudet Dictionary of American Sign Language [60], performed by native signers. Each sign is annotated with the gloss label (approximate English translation), start and end frames, hand shapes at the start and

end frames, and position of the hands and face, with multiple examples per sign. Such datasets, while useful, lack any information about scene depth, since they were recorded with standard RGB cameras. Thus, when using them, researchers suffer from the limitations of having to use conventional 2D hand detection and tracking algorithms based on, for example, skin color and motion.

There are also available some 3D gesture datasets. Guyon, et al., present a Kinect–based 3D gesture dataset for the ChaLearn gesture recognition competition in [21] that contains 50,000 gestures recorded by 20 different users, organized into 500 batches of 100 gestures. Compared to the ChaLearn dataset, the dataset described in this chapter has certain advantages. First, our dataset is recorded at a higher frame rate of 25 frames per second (fps), as opposed to 10 fps. Secondly, only 400 frames are manually annotated with any skeletal information in the ChaLearn dataset, which makes it difficult to quantify the efficacy of any body part locator or tracker being developed. Our dataset contains skeletal information for every frame of each sign. Third, our dataset sonsists of a much larger vocabulary. Finally, as the ChaLearn videos are offered only as AVI files, we cannot translate the pixels into x, y, z coordinates in a 3D world reference frame. Our dataset provides access to the raw scene depth information and allows us to determine the x, y, z coordinates in the Kinect reference frame.

The MSRGesture3D dataset from Zicheng Liu at Microsoft Research is an ASL dataset recorded with a Kinect [61]. It consists of 12 dynamic signs from 10 signers, each performed 2-3 times, with the hands segmented above the wrist. Due to its limited vocabulary size, it is unusable in this work.

The most similar dataset to ours is the large DEVISIGN database of standard Chinese Sign Language created in a collaboration between Microsoft Research Asia, the Chinese Academy of Sciences, and Beijing Union University recorded by 30 signers

[44]. This dataset is particularly useful for recognition methods, such as HMMs, that require a larger dataset for training. While it has a larger vocabulary than ours and consists of more signers and training examples, it only provides Kinect annotations for the joint positions. It lacks the manual annotations of body part locations that can provide the ground truth for evaluating, for example, hand detectors, as well as any temporal segmentation of the signs.

4.3 Dataset

The goal is to create a structured motion dataset that will enable researchers to explore body part detection and tracking methods, as well as gesture and sign recognition algorithms not possible with such datasets as the ASLLVD [62] by using depth-based features. The dataset is being recorded with a Microsoft Kinect. Figure 4.1 shows an example from one of the recording sessions. In the depth image of this figure, the darker gray areas of the image are located closer to the camera. The black regions are portions of the scene for which depth information was not available in the IR shadows.

4.3.1 Size and Scope

Ultimately, the final dataset will contain the 3,000 signs found in The Gallaudet Dictionary of American Sign Language [60], offering an abundance of complex movements of the hands and arms. Currently, 1,113 signs, both one-handed and two-handed, have been recorded with one fluent signer and 750 with another, but we are in the planning stage to record additional signers, so that there are multiple examples of each sign. As with [59], fingerspelled signs, loan signs, and classifiers are not included in the dataset. A fingerspelled sign is a word that is spelled out by using the manual alphabet. When a signer has to use a letter that is part of the overall

sign, that letter is known as a *loan sign*. Classifiers provide additional information about the object being signed, but since there are infinite variations of them, they are excluded.

4.3.2 Technical Specifications

Both the Kinect color frames and depth frames have a resolution of 640×480 pixels and are recorded at frame rate of 25 frames per second. Since the native resolution of the Kinect depth sensor is 320×240, the depth frames are resized using interpolation.

The signers perform groups of ten signs per video in front of a plain backdrop in a lab with consistent lighting. The signs are performed while standing, and the scene is framed so as to include the region from about the knees to several inches above the signer's head. This ensures that the entire signing space is included in the video frames. Each video begins with a calibration pose that can be used to detect the signer and initialize tracking. After the pose, between each sign, and after the last sign, the signer returns her hands to her side, ensuring a clear separation of the signs in the video.

The now defunct PrimeSense, LTD. OpenNI framework [63] was used to record the signs in the ONI format. OpenNI was an open source sensing development framework used in many third party APIs. Its purpose was to standardize compatibility and interoperability of Natural Interactive devices and applications. Though the company no longer exists, OpenNI and third party software developed around it are still useful to and are used by researchers that want to develop their own detection and tracking tools. Compressed and uncompressed 8-bit AVI videos of the recordings are also available for both the RGB and depth sensors, as are 16-bit binary files containing full resolution depth data.

All new signers, however, will be recorded with the improved time-of-flight-based Microsoft Kinect version 2, a device with higher resolution in both depth and RGB video, more precise depth measurements, and an improved depth range. The new skeleton joint positions will be provided by the Microsoft Kinect SDK, and preliminary experimentation—described in Chapter 6—has shown the joint locations to be more stable and reliable than those from NiTE.

4.3.3 Annotations

Each video in the dataset is annotated with the start and end frames of each sign so that any sign can be quickly accessed. The first depth video frame of each sign is manually annotated with a bounding box around the signer's face to give an idea of the scale of the individual in the video. Using this information, the researcher can scale the query and model signs to be the same size before comparison. Furthermore, every depth frame belonging to a sign is manually annotated with bounding boxes around the hands. The hand and face annotations for an example sign frame are shown overlain on the depth frame image in figure 4.2. In addition to the position and frame information, the annotations include information about the signs themselves, such as signer ID, file locations, sign type (two-handed or one-handed), and gloss, or approximate English equivalent.

4.4 Hand Detection Benchmark

In order to establish the benchmark, we chose to use the hand location capabilities of the user skeleton tracker included in the OpenNI 1.5 NiUserTracker sample program [63], since it was freely available to everyone. In particular, the upper body joint positions provided by the tracker were recorded for each frame of the videos, as well as the confidence level of those positions.

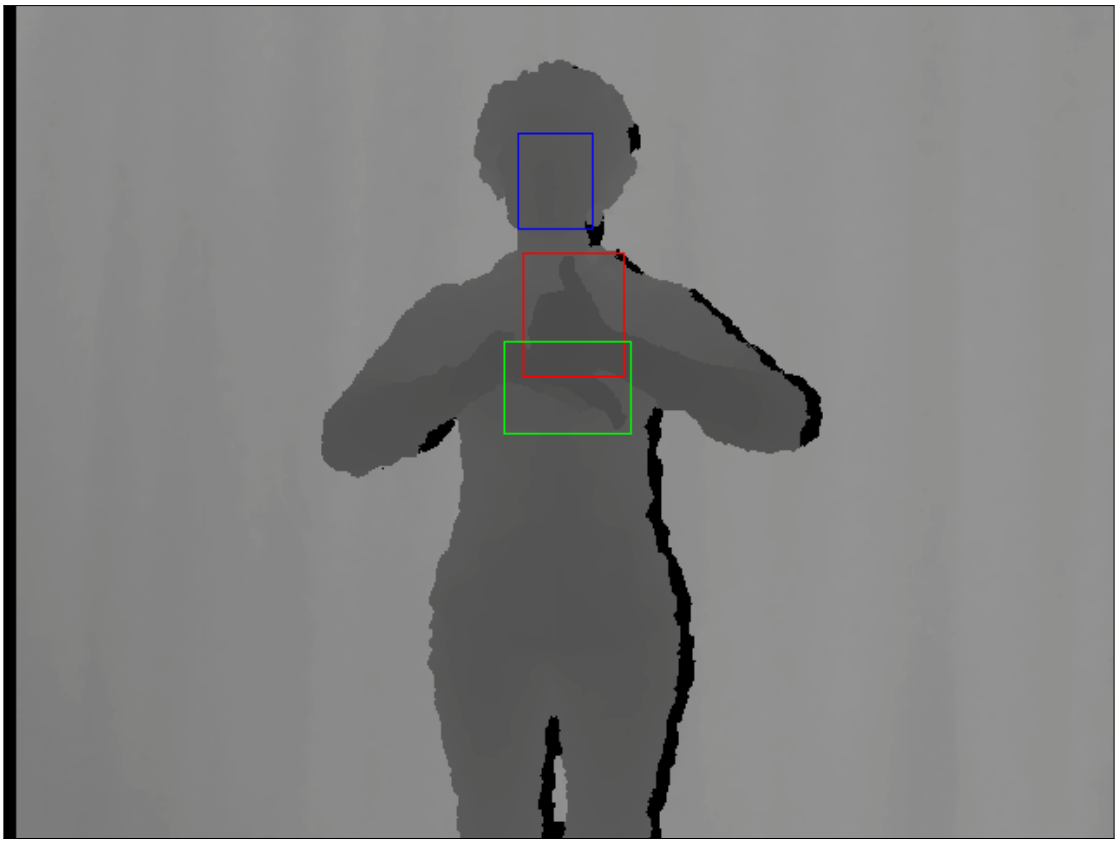


Figure 4.2: Sample hands and face annotations of a single depth video frame.

To evaluate the efficacy of using the skeleton tracker to approximate the positions of the hands, we used 606 signs of varying complexity from the dataset—206 one-handed and 400 two-handed—and processed them with the tracker. For one-handed signs, only the signing hand was considered. Once the hand positions were obtained, they were compared to the ground truth positions from the manual annotations, and the Euclidean pixel distance between them was recorded as a score, so that a lower score would indicate a closer estimation of the hand's actual location. This operation was performed on each frame of the signs, and the accuracy was calculated to serve as the benchmark for the evaluation of future methods.

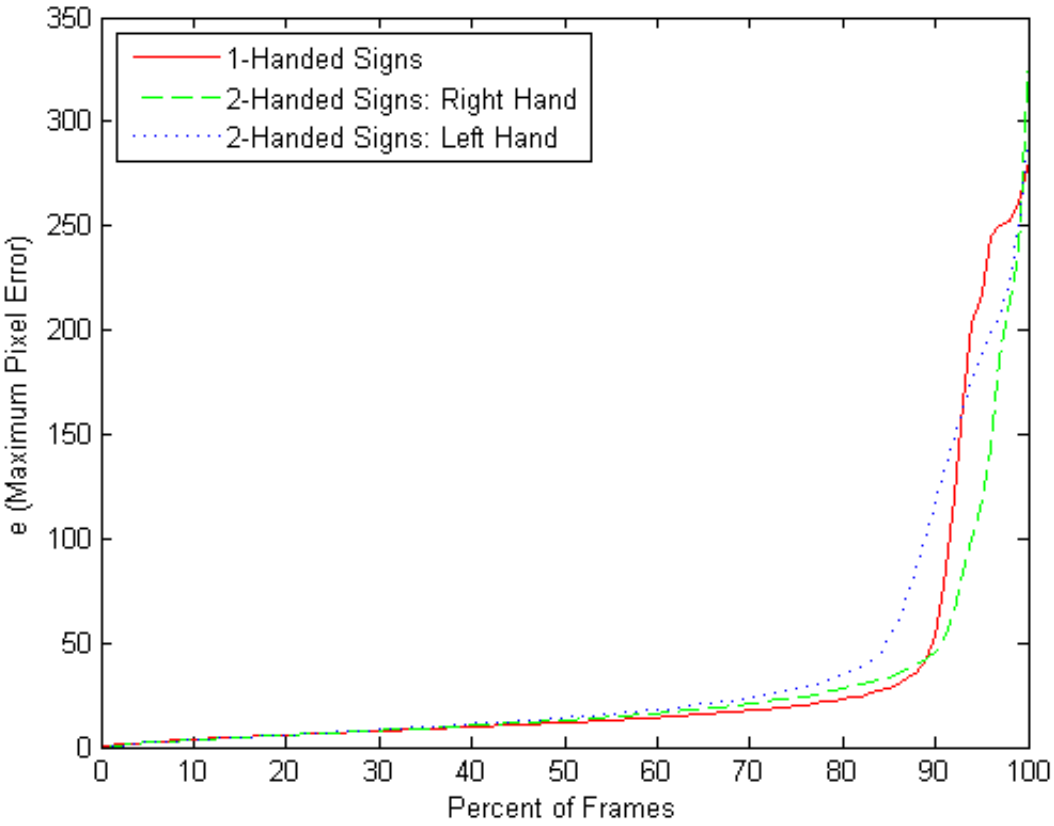


Figure 4.3: Skeletal tracker pixel error for hand locations.

Accuracy is described as a percentage of frames in which the automatically generated hand locations fell within in various pixel distances (termed pixel error) of the manual hand annotations. Figure 4.3 shows the accuracy of the skeletal tracker in locating the signer's hands in both one-handed and two-handed signs. For example, in 80% of the frames of two-handed signs, the skeletal tracker had a pixel error of 27 pixels or less for the right hand. These results set the standard to which the hand detection method proposed in chapter 6 will be compared.

We also calculated the maximum pixel error for each sign, separated into one-handed and two-handed signs. Figures 4.4 and 4.5 show the results for the skeletal tracker and its comparison to the single hand detector, respectively. We can see in

figure 4.5, for example, that 50% of the signs had a maximum pixel error of about 22 pixels or less when the comparison method of [20] was used to detect the hands.

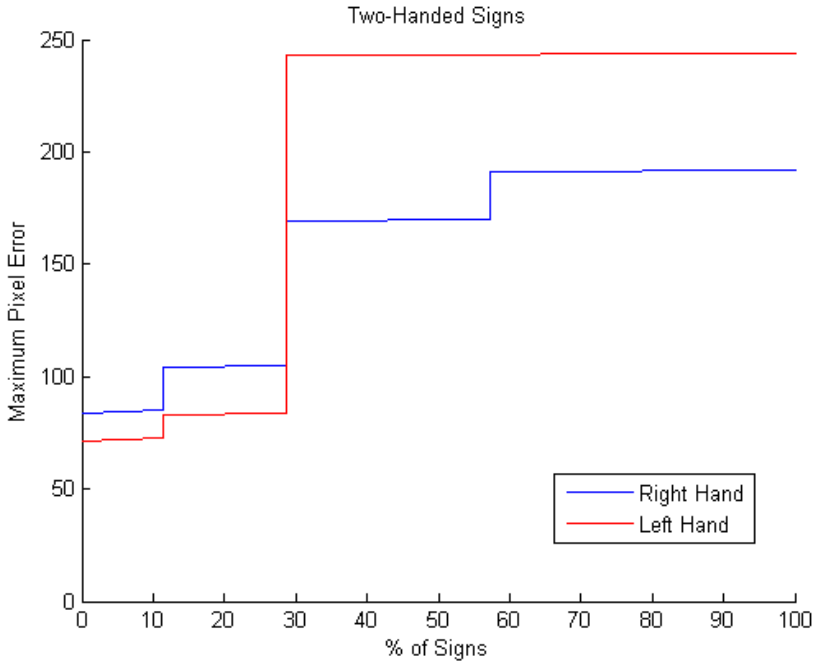


Figure 4.4: Maximum hand location pixel error on a per sign basis for the skeletal tracker.

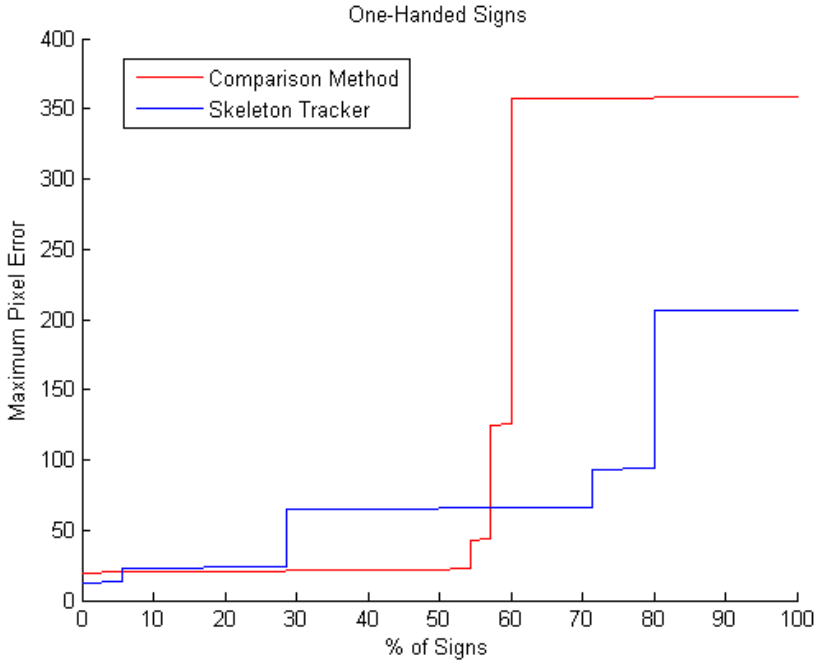


Figure 4.5: Skeletal tracker and one hand gesture method maximum pixel error on a per sign basis.

CHAPTER 5

BASELINE ASL RECOGNITION EXPERIMENTS

5.1 Introduction

In Chapter 4, a 3D structured motion dataset was presented that can be used to test body part detection and gesture or ASL recognition algorithms. The chapter also presented a hand detection benchmark to which the methods developed in this thesis can be compared. This chapter deals with ASL sign recognition, and several methods are proposed for improving recognition rates. Section 3.1 introduces the similarity measure on which the methods developed in this thesis are based. Using this similarity measure, Section5.2 establishes a sign recognition benchmark that provides results from using both the ground truth hand location knowledge obtained from manual annotations and the imperfect hand location knowledge from a popular Kinect skeleton detection algorithm. This allows comparisons of potential improvements in both the sign recognition and hand detection methods. The standard set forth in this chapter will be used throughout this thesis to evaluate the efficacy of the proposed methods.

5.2 Sign Recognition Benchmark

This section introduces the benchmark we created for evaluation of the methods presented in this thesis. In it, we describe the model and test sets used for the experiments, as well as the experiments, and provides a discussion of the results. This discussion outlines problems inherent in large vocabulary gesture sets and Kinect provided hand positions that reduce recognition accuracy.

5.2.1 Training Set

Due to the lack of publicly available 3D ASL datasets and the incomplete status of that described in Chapter 4, we chose to use a standard 2D RGB dataset for training and were, thus, working with 2D projections of the signs onto the image plane. In the experiments, we used a 1,113 gesture vocabulary training set to which we matched our smaller subset of RGB-D signs. To ensure user independence, no videos from the test set signer appear in any of the training sets. As additional annotations are made available and the 3D dataset is expanded to include several additional signers, 3D trajectory matching experiments will be performed to establish an additional benchmark. Three examples each, from different signers, of the 1,113 gesture vocabulary were taken from the dataset described in [59] to be used as training examples. Since the videos are standard 2D RGB videos and there is no real-world distance information for the hand and head positions, we used their pixel locations in our training data.

5.2.2 Test Set

We calculate baseline accuracy on two datasets, *JK850* and *CK368*, from our 3D dataset recorded with OpenNI [63]. The first set consists of a mix of 850 one-handed and two-handed signs of varying complexity. The second consists of 368 signs and is a more difficult set due to the wide variation in performance from the models. We used the NiTE skeleton tracker [64] to determine the hand positions in each frame and the head position in the first frame of each sign. As the NiTE tracker provides positions for joints in a 3D Kinect-centric coordinate system, we used the projections of those positions onto the 2D depth image plane, so that instead of the real-world distance measures for the joints, we were using their pixel coordinates. This allowed for proper comparison with the pixel coordinates used in the training set, once

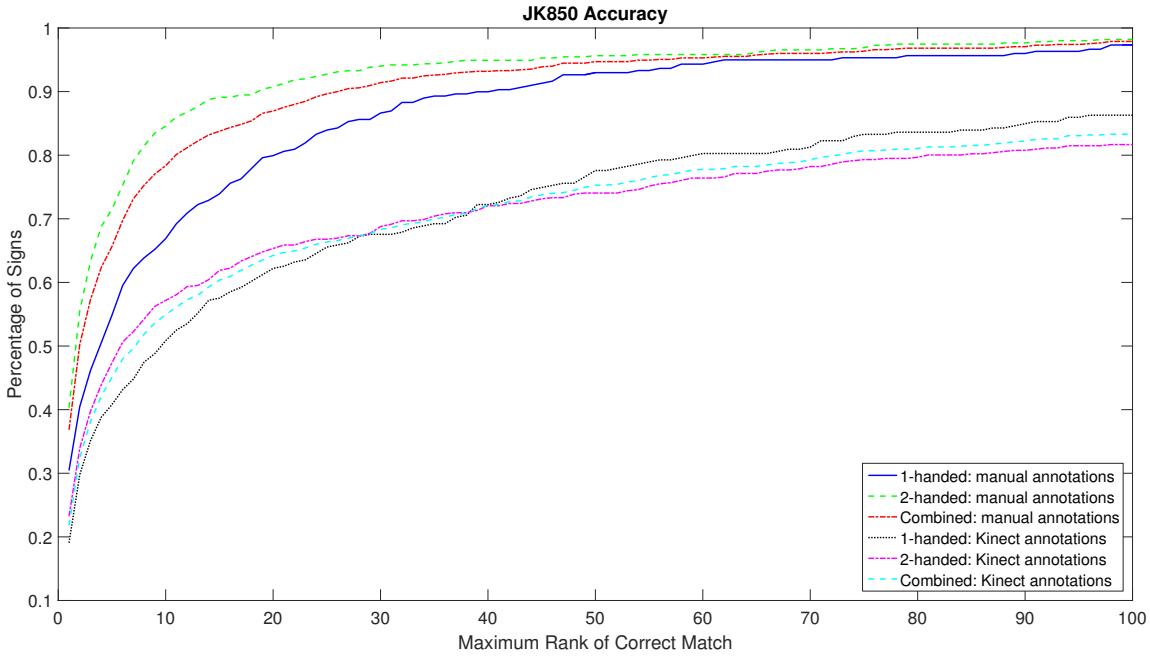


Figure 5.1: Baseline accuracy for JK850

they were expressed in the face-centric coordinate system and normalized to ensure translation and scale invariance. For comparison to a best possible scenario using this similarity measure, experiments were performed using the manual annotations of the hand positions in each frame in addition to the skeleton tracker generated positions.

5.2.3 Benchmark Results

After running the gesture recognition experiments using the skeleton tracker data, the accuracy was calculated as a percentage of signs for which the correct match was ranked in the top k results returned by the system. Figures 5.1 and 5.2 show the results on the individual datasets. For example, 65.3% of the two-handed signs in the *JK850* dataset ranked in the top 20 matches using the Kinect annotations vs. 55.8% of the *CK368* set.

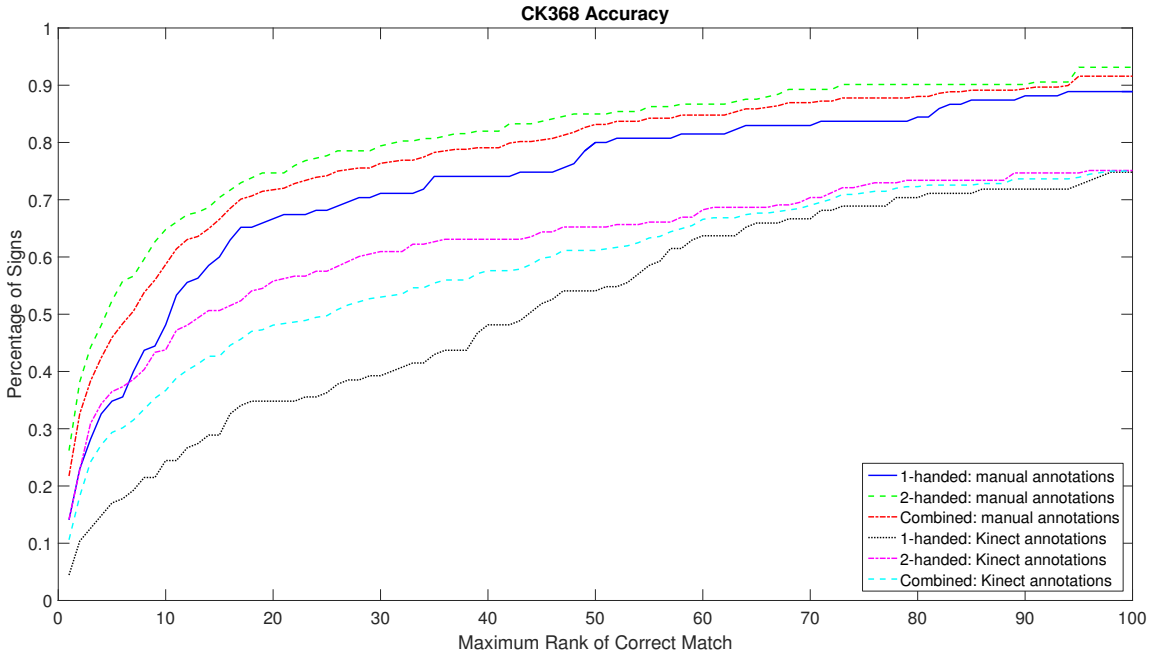


Figure 5.2: Baseline accuracy for CK368

Figure 5.3 compares the combined one-handed and two-handed accuracy of the *JK850* and *CK368* datasets. The results indicate that the *CK368* dataset is a more difficult set.

Finally, we present the overall accuracy on all signs using the skeleton tracker and manual annotations in figure 5.4, giving both datasets equal representation in the results. This figure gives us two goals. First, both the best-case and skeleton detector results provide a benchmark for improvement of the similarity measures by incorporating the methods discussed in Sections 7.3 and 7.4. Second, the results of experiments using Kinect skeleton output provide a baseline for comparison of future automatic hand detection methods being developed. There is a clear discrepancy in the accuracy using the manual annotations and Kinect annotations. Since automation is an ease-of-use requirement of this ASL dictionary system, a major goal becomes to improve recognition using Kinect annotations. The methods presented in sections

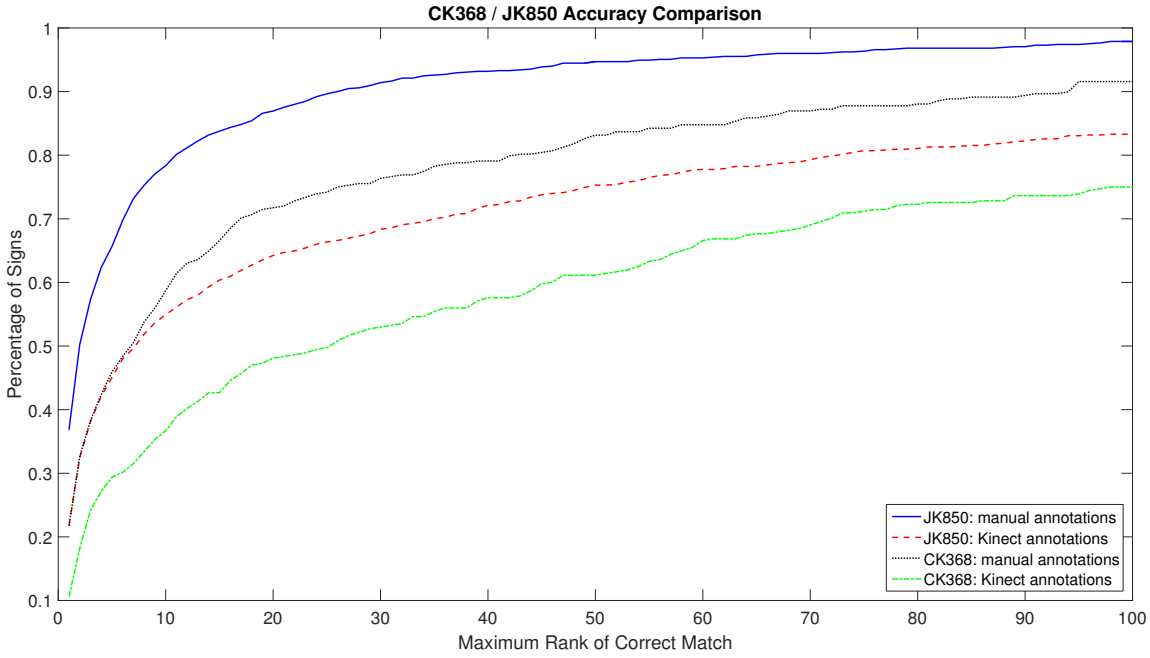


Figure 5.3: Comparison of JK850 and CK368

7.3 and 7.4 work toward this end and show a significant improvement on signs using the Kinect annotations.

5.2.4 Discussion

There are several key observations to be made when examining the results. One is that two-handed signs tend to be easier to recognize than one-handed signs. This is due to the increased uniqueness of the sign classes resulting from having trajectories for two hands. When there is only one trajectory to compare, many signs will share similar trajectories, especially since we have many static signs. One of the goals in this research thus becomes to address this discrepancy. Chapters 7.3 and 7.4 address this concern to an certain extent, and do significantly improve one-handed accuracy, but it is left to future work to create classifiers that are specific to one-handed signs. DTW is not necessarily the best method to use.

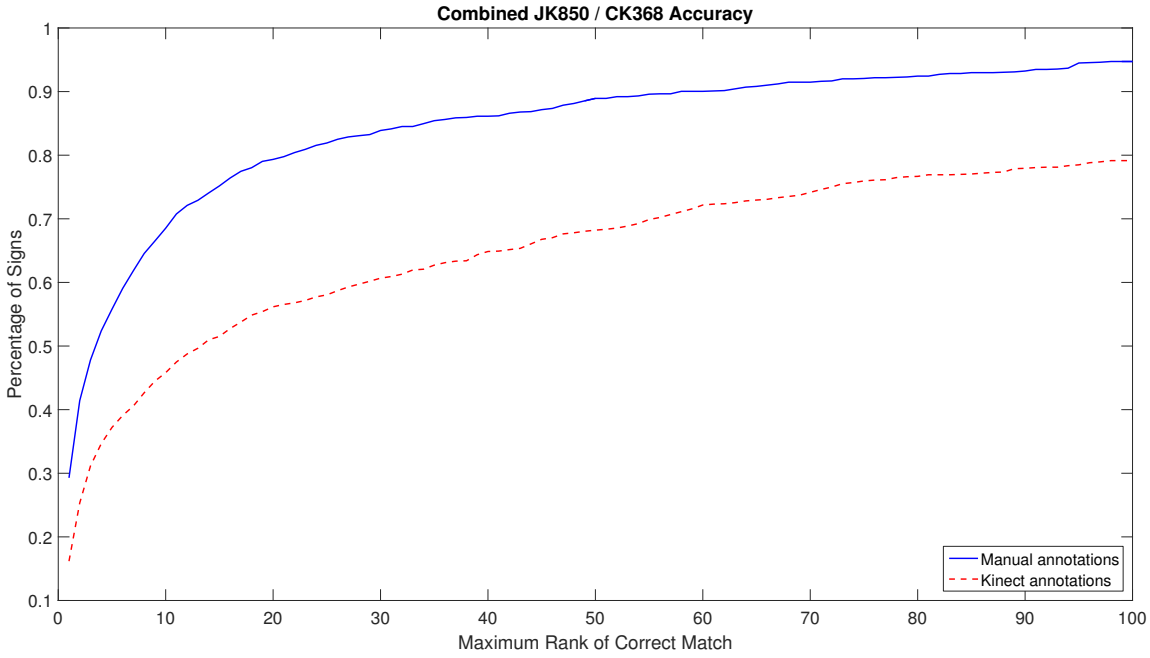


Figure 5.4: Combined Accuracy of JK850 and CK368

A second key observation to make is the large difference in accuracy between the two datasets. It is clear that the DTW-based recognition struggles on the *CK368* dataset. This is due to the fact the its signer produces signs with different degrees of gesture exaggeration when compared to our model sets, resulting in much different sizes of trajectories with far more warping of the sequence than DTW can sufficiently handle. There tends to be a significant degree of trajectory misalignment that causes incorrect matches. Chapters 7.3 and 7.4 do a great deal to account for these variations in gesture size and expressiveness by modeling these variations among same-class signs and by both refocusing the coordinate system on different parts of the trajectory and adjusting the manner in which a sign is size normalized. The results of the experiments in Chapter 8 show a far better improvement in accuracy on the *CK368* dataset and demonstrate the potential of the methods in a dictionary system where students of

ASL will likely perform the query signs with large variations compared to the example signs.

When we examine the results and take note of the specific signs with poor recognition rates, two other general causes of problems become apparent: large vocabulary sign similarity and skeleton tracker inaccuracies. The intra-class variation modeling and multiple-pass dynamic time warping proposed in sections 7.3 and 7.4 also help minimize the impact of these problems.

5.2.4.1 Gesture Similarity

Many gestures, particularly one-handed gestures, share a similar trajectory, and it can be seen in figures 5.1 and 5.2 that the one-handed signs are matched in the top 10 signs at a much lower rate than the two-handed gestures. Many of these signs are static gestures, in which the position is approximately the same across signs and only hand shape differs. Figure 5.5 shows frames from 4 such signs.

It is evident that the skeleton tracker alone does not output enough information to distinguish between the signs, since it does not estimate the structure or finger configuration of the hand itself. In order to incorporate automatic hand shape detection, this would first require the development of an algorithm to cluster the pixels belonging to the hands using the Kinect hand positions as a starting point. The positions, however are far too unstable to use them directly, and this clustering is beyond the scope of this work. The incorporation of hand shape or appearance comparison, however, can significantly improve the results and is left for future work. The methods presented in this thesis instead try to account for the variation in position and size of the performed signs, as well as the arbitrary trajectory resulting from the inability of a signer to keep the hand still in static signs.

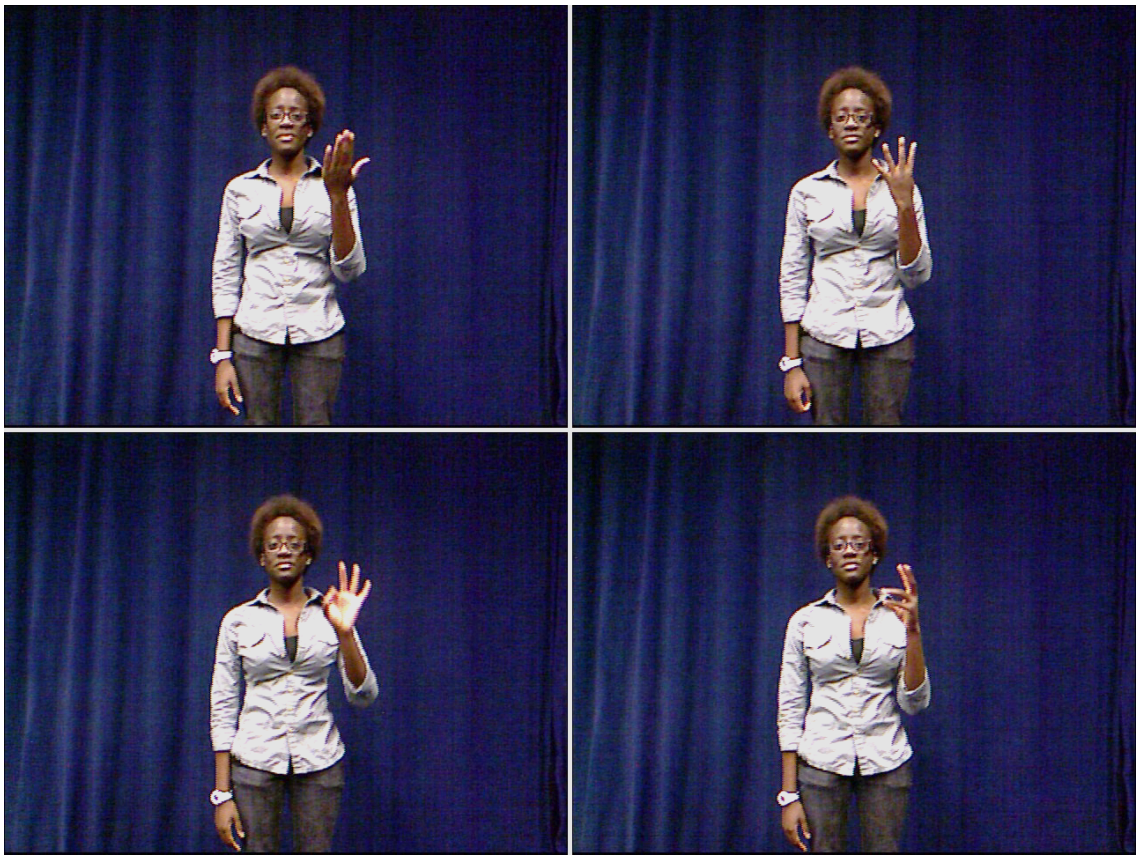


Figure 5.5: Examples of sign similarity. The position is roughly the same, but the hand shape differs.

5.2.4.2 Skeleton Tracker Inaccuracies

It is clear that existing skeleton trackers are not designed for tracking complex and intricate skeletal joint movement. Joint proximity to each other or the body can cause problems. The current depth-based trackers sometimes fail in instances when the hands and arms come into contact with the body, likely due to the limited depth resolution of the Kinect, as well as limitations inherent in the depth disparity feature that is used by the algorithms [65]. Signs for which there is no clear separation and obvious distance between the limbs and the body often cause the tracker to lose the joints.

In the test dataset, when the signer lowers her hands between signs and places her arms at her sides, the tracker often loses lock of the joints as they blend into the mass of the body. When she lifts her arms to perform the next sign, the tracker can take a significant portion of the sign to relocate the joints. Such is the case in figure 5.6. The green shows the centroid of the manually annotated bounding box, while the red shows the skeleton tracker hand estimate before it reacquired the hand position. This would indicate that the NiTE skeleton tracker is attempting to track a located joint rather than detect it in each frame as the Microsoft Kinect SDK detector does. This may not be an issue in a sign language video dictionary system when the user

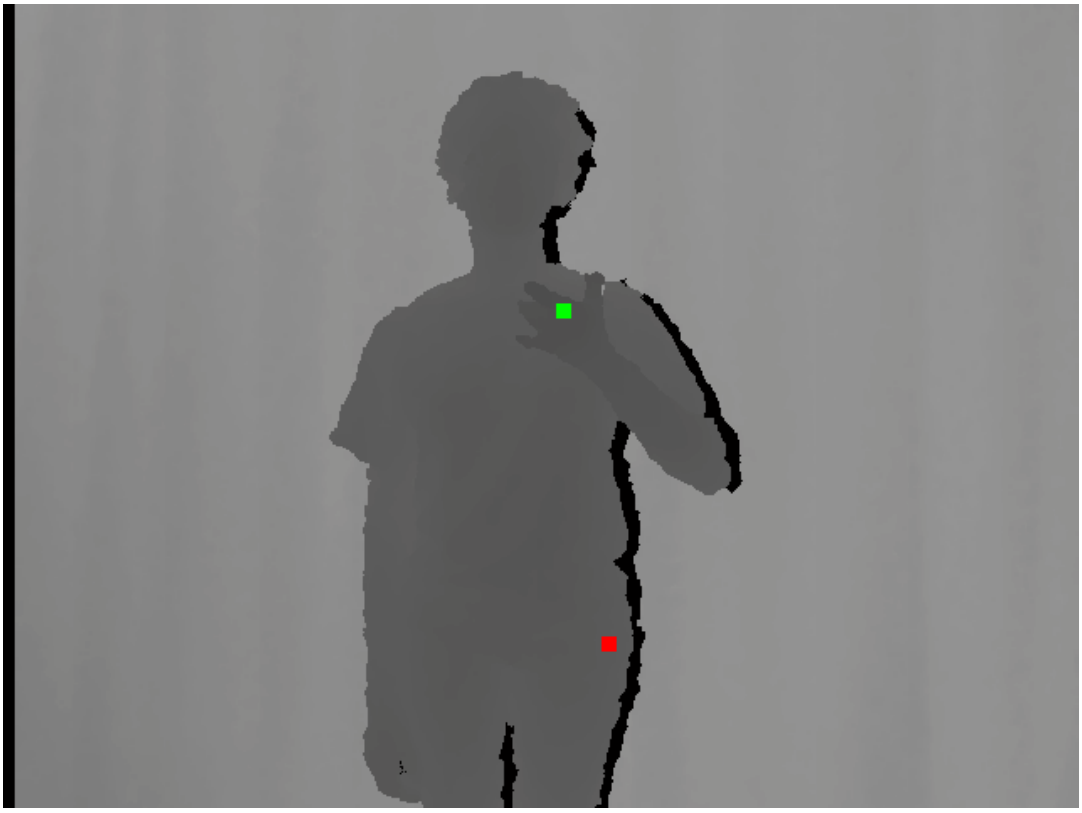


Figure 5.6: Failure of the skeleton tracker after the signer's arms were at her side. The red square is the tracker hand position estimate. The green square is the centroid of the hand bounding box.

can ensure that the tracker is properly tracking movements before performing the sign.

Gestures in which the arms cross can also provide considerable difficulty for the skeleton tracker. The tracker struggles to distinguishing between the arms, the joint position estimates begin to destabilize, and the tracker loses lock on the joints.

There are also joint estimate stabilization issues between frames. Even when the skeleton tracker does not lose track of the joints, the hand position, for example, can jump around the hand from frame to frame, even in a static gesture in which there is little movement of the hands. When part of the feature vector extracted from hand positions includes various directions of motion and changes in those directions from frame to frame, this instability can have an effect on scores and recognition accuracy. The newer Microsoft SDK appears to address this issue to some extent, but it remains a problem. Work is clearly needed on the stability of joint position estimates and the responsiveness of tracking to movement, but it is beyond the scope of this work.

It is evident that the existing trackers are geared more towards whole body pose estimation and do well in recognizing action poses that use large deliberate movement, such as kicking, jumping, large arm movements, etc [65]. This makes sense, as the Kinect was designed to be part of a gaming system. Besides these full body poses, the only hand gestures it was designed to handle are for simple menu navigation. Due to the depth disparity feature used, the Kinect was never intended to achieve fine-grained discrimination between joints in close proximity to the body.

CHAPTER 6

ASL VIDEO DICTIONARY SYSTEM

6.1 Introduction

The ASL Video Dictionary System (ASLVDS) presented herein offers users the ability to quickly and easily search for the meaning of an unknown sign in a more integrated and automated manner than the previous variant, while improving accuracy and reducing the time required per query. It eliminates inconsistencies in sign size normalization due to human factors and streamlines the dictionary search process.

There are several advantages over the previous system. The previous system required the use of two pieces of software. The user would first record a video of the sign using third-party web-cam software and would then import the recorded video into the dictionary system for sign matching. The system described in this chapter integrates the recording and matching into a single program, thus making sign search both easier and faster.

The earlier version of the dictionary system also required the user to initialize a hand tracker and trajectory generation algorithm by drawing bounding boxes around the hands and face in the first frame of the sign. The tracker could then track the hands throughout the sign. If the tracker lost the hands during the sign, the user could then correct the frame where it lost track and have the algorithm retrack the hands. This requirement is eliminated by using a *Microsoft Kinect for Windows* v2 [66] to automate the hand detection process, but other methods may also be employed.

As the recorded query signs are scaled based on the size of the user's face in the previous system, differences in the sizes of the bounding box users draw for the face can affect system performance. This system seeks to eliminate inconsistencies by using a proportion of the distance between two easily located joints in the user's skeleton as detected by the RGB-D skeleton detection algorithm. This proportion is learned through experimentation on a validation set containing none of the users that participated in this study.

The system is evaluated by performing a series of sign match accuracy and timing tests on a random set of signs from the 1,113 sign vocabulary, employing a user-independent experimental protocol. In order to recreate a realistic usage scenario in the tests, participants with little to no knowledge of ASL are used, and none of them are familiar with the dictionary system itself.

6.2 Video Dictionary-Specific Related Work

Recent work using RGB videos for sign language recognition is found in [67, 68, 69]. Bragg [67] proposes a system called ASL-Search. When a user encounters an unfamiliar sign, ASL-Search requests the user to select sets of features including hand shapes, orientations, locations, and movements from the interface based on his/her observation, and recognition is based on these selected features. Requiring the user to select from among these features reduces system ease of use, and can introduce potential for error, as the user may classify a feature incorrectly.

Depth sensing technology, like the Kinect, has also been explored for sign language recognition systems [70, 71]. Elliott et al. [71] proposed a Kinect camera based sign look-up tool which includes an interactive sign recognition system and a real-time sign synthesis system. The method uses hidden Markov models (HMMs) to model signs. Pavlakos et al. [70] combined visual cues (color and depth images) and au-

dio under an HMM framework, and the proposed method was evaluated on a public gesture dataset: the ChaLearn multi-modal gesture challenge dataset [21]. Given the amount of training data often required by such techniques, this is an unreasonable approach for this dictionary system.

6.3 The ASL Video Dictionary System

This section introduces the new fully integrated RGB-D ASL video dictionary system and experiment platform for which the methods in this thesis are being developed, and outlines the improvements over the previous system. It demonstrates an improved sign match accuracy and significantly reduced time required per query.

6.3.1 System Description

The ASL Video Dictionary System is a combination of hardware and software: Microsoft's Kinect 2 RGB-D sensor and a custom Graphical User Interface (GUI). The system is written in C++, using the Qt 5.3 application and UI framework [72] for the GUI, OpenCV 2.4.9 [73] for image processing, and Microsoft's Kinect SDK v2 to access the sensors and generate the skeleton data [74]. At present, the dictionary is trained three examples each of the 1,113 sign vocabulary discussed in chapter 4, obtained from the American Sign Language Lexicon Video Dataset [75].

6.3.1.1 Hardware

The dictionary system uses a Microsoft Kinect v2 RGB-D sensor and the associated Kinect SDK v2 to provide several streams of data. Specifically, the depth, color, body, and body index streams are utilized. Whereas the color (RGB) stream provides standard 1920×1080 pixel video frames, the depth (D) stream provides scene depth information with a resolution of 512×424 pixels. It is this scene depth information

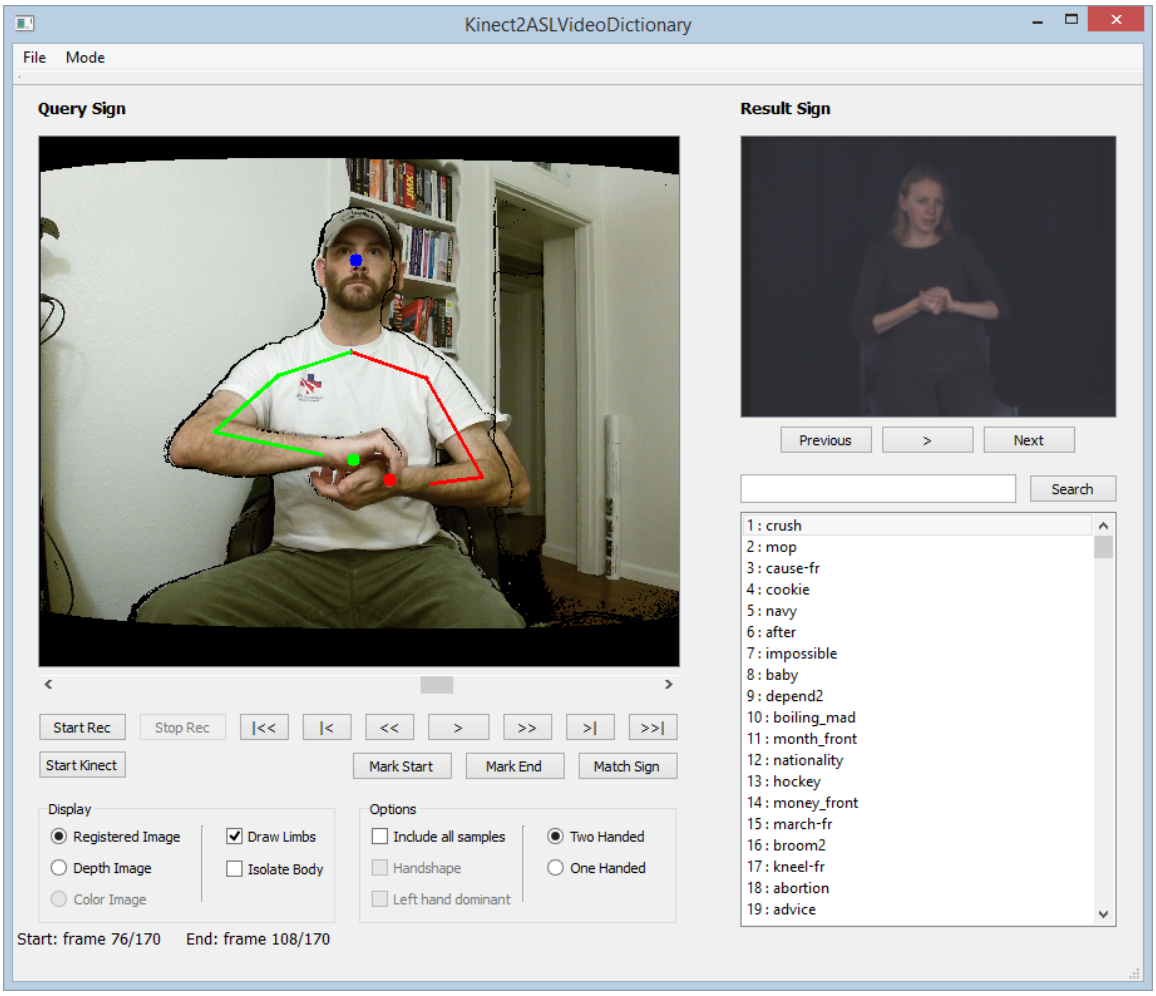


Figure 6.1: The ASL Video Dictionary System

that enables the skeleton detector to work (see [65] for details). The output of the skeleton detector is provided by the body stream, which consists of the 3D coordinates (with the sensor as the frame of reference origin) of all 25 joint positions that the Kinect SDK v2 provides. The coordinates can be mapped to their corresponding 2D projected pixel positions in the depth image. The body index stream provides information about which depth frame pixels belong to different people in the video. Example corresponding color, depth, and registered (color pixels mapped to depth pixels) video frames can be seen in figure 6.2.

(a) Color Frame

(b) Depth Frame

(c) Registered Frame

Figure 6.2: Example corresponding color, depth, and registered video frames

6.3.1.2 Graphical User Interface

The system GUI is composed of two main sections: a query recording section and a results section. The query recording section allows a user to select a video stream to display, record and review a video, ensure skeleton detection works properly, perform temporal segmentation of the sign, and initiate the matching process. When the user records a video, all data streams are recorded, though only one is displayed.

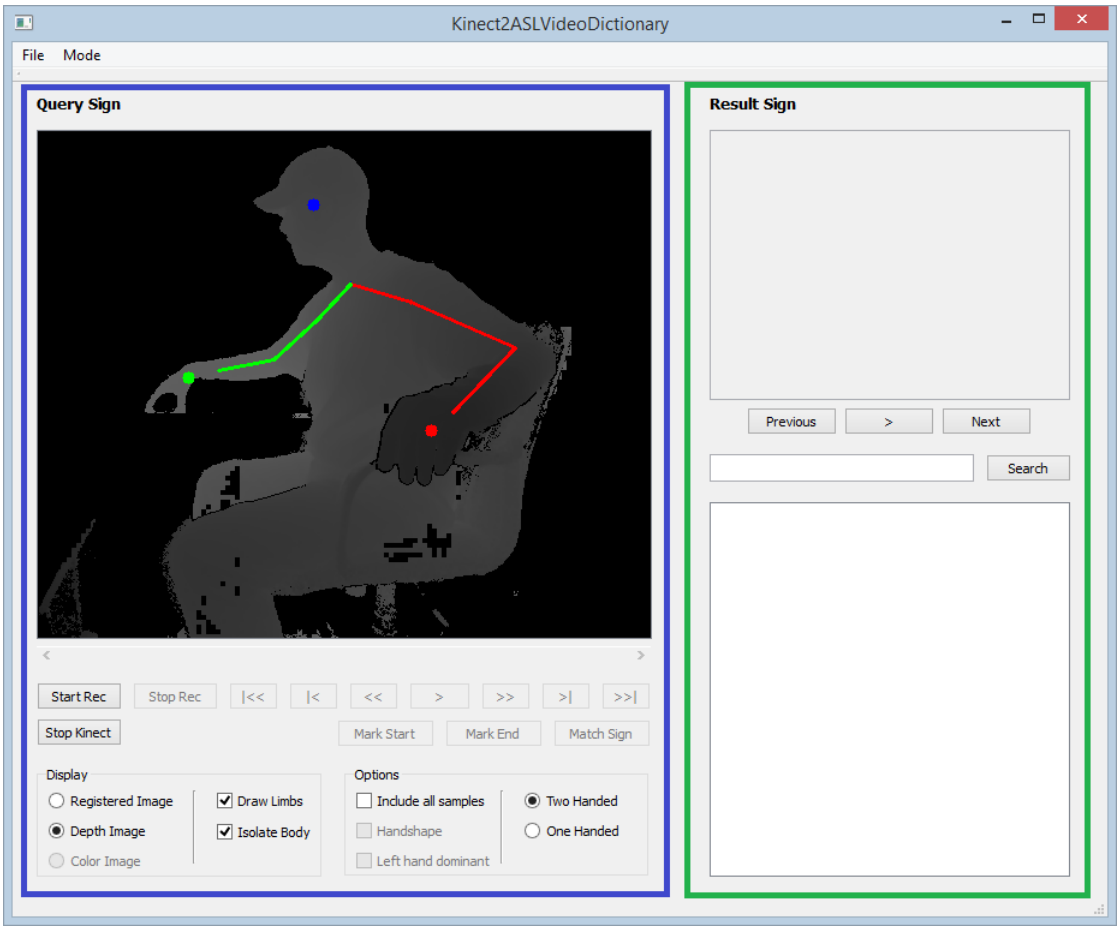


Figure 6.3: The system GUI: Highlighted in blue is the query recording portion. Highlighted in green is the results section.

The results section presents a ranked list of sign matches, enables the user to view a video of each matched sign, and provides result search capabilities. Figure 6.3 shows the GUI before recording of a sign has begun. The depth stream has been selected, the user's body isolated, and the limbs are being drawn. Figure 6.1 shows the GUI after a search has been performed. In it, the registered image—in which each depth image pixel's corresponding color pixel is drawn—is displayed with the head and hands positions and limbs drawn.

6.3.1.3 Feature and Trajectory Generation

The hand positions are expressed in the new coordinate system using the skeleton detector head position in the first frame of the sign as the origin, as described in Section 3.2. Instead of using the face size, the sign size is normalized based on a proportion of the head-neck joint distance; this proportion was learned through experimentation. The new coordinate system and resizing ensures scale- and translation-invariance. Future experimentation will determine if the shoulder-to-shoulder distance would be a more appropriate basis for resizing the sign. Preliminary experiments show that this might be the case, as shoulder width, rather than head-neck distance, can be a better indication of both the size of and the arm length of the user.

6.3.1.4 Sign Matching

Using the method described in Section 3.1, two sign rankings are generated. One contains all three examples of each sign of the same type, while the other contains just the best scoring of the three examples in each sign class. Either set of rankings can be displayed.

6.3.1.5 Results Display

When sign comparison is complete and the results are ranked according to similarity, a list of the definitions of the results is generated and displayed. By clicking on any of the results the user can play a video of the sign to determine if it is correct match. Since there are three training examples of each sign and two ranked lists are generated as described in Section 3.1, the results display output mode can be selected to show either all matches or just the matches using the lowest of the three scores from the training examples. The results section of the GUI also provides search

functionality, which can be useful in an experimental setting in which the meaning of the sign is known.

6.3.1.6 System Usage

There are three phases to using the system. First a query sign is recorded. Second, temporal segmentation is performed to isolate the sign. Third, handedness is selected, matching is performed and the results displayed.

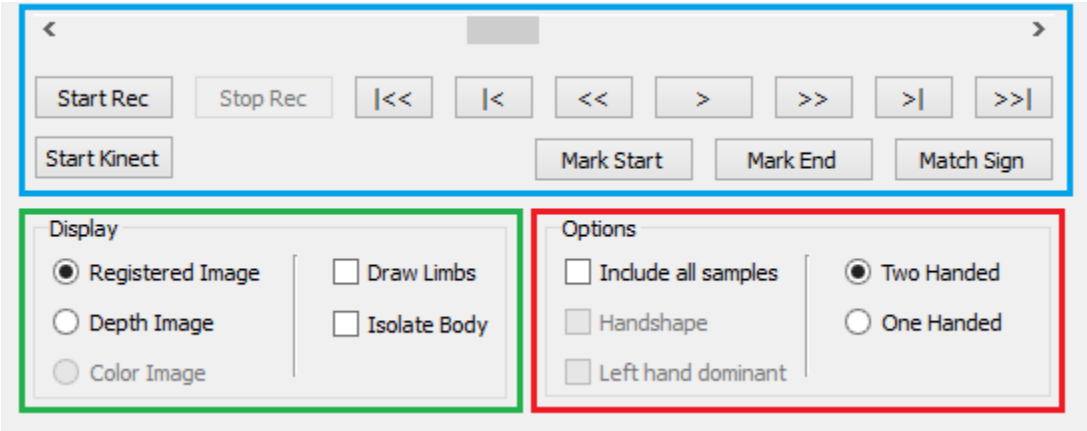


Figure 6.4: System Controls

When the ASL Video Dictionary System starts, the Kinect starts streaming and the live video is displayed along with the locations of the hands and head; the user has the option of adding limbs to the display. When ready, the user presses the record button, and the system starts recording joint positions (both 3D world coordinates and 2D pixel locations), joint states (i.e. tracked, inferred, or not tracked), the 16-bit depth video, the color video, and the registered video, in which each depth pixel is mapped to its corresponding color pixel. The left side of the blue outlined area in figure 6.4 contains recording and streaming controls that allow the user to start and stop the Kinect as well as the query recording. The green area of figure 6.4 contains

the available data stream and display options. All streams, as well as timing and accuracy data, can be written to disk for later analysis and detection of bottlenecks.

The user then performs the sign and can use the skeletal overlay on the display to ensure the detector is working properly. When the sign has been performed to the user's liking, he can stop the recording, view what has been saved, and move on to temporal segmentation. The slider and buttons in the top and right sections of the blue area in figure 6.4 allow the user to set the start and end frames of the query and to play the full or cropped recording with the skeletal overlay.

During the temporal segmentation phase, the user marks what he or she feels are the start and end frames of the recorded sign. The user can again review the segmented sign and make any changes to the start and end frames that may be needed. After ensuring that the appropriate sign type (i.e. one or two handed sign) has been selected in the matching options section (red area of figure 6.4) and whether it is left hand dominant in the case of two-handed signs, the user can begin the match process.

When the *Match Sign* button is clicked, the joint position data is cropped and sent to the trajectory generation algorithm, which builds the modified version of the feature vectors described in Section 3.2. As the skeleton detector does not provide a face bounding box, the system uses a portion of the head–neck distance to normalize the size of the sign. This portion was learned through experimentation on a signer not participating in the study. The feature vectors are then used for sign matching using DTW, as described in Section 3.1. After matching is complete and the results displayed, the user looks through the videos associated with the ranked list for the sign in question.

6.3.2 Experiments

An assortment of experiments were run to evaluate the performance of the system with respect to sign match accuracy and the amount of time it takes to use the system. To simulate a typical usage scenario, we chose experiment participants with little to no experience with sign languages. This enables us to asses the performance of the system with a typical user, instead of a regular user of ASL who knows how to perform the signs properly. Furthermore, they had never used the system before and had not yet developed the ability to very quickly perform a sign search.

For the study, five participants, designated **P01**–**P05**, were given a brief introduction to the system so they could observe how to use it and were presented with video examples of 30 signs chosen randomly from the 1,113 signs in the dictionary system's vocabulary. A separate set of 30 random signs was generated for each participant. After viewing video of the sign to be performed, the participant used the system to search for the meaning of the sign without intervention from the experiment coordinators.

For each sign that the participants performed, the color, depth, and registered videos were written to disk, as well as position information for all 25 joints output by the skeleton detector, the ranked results lists, and timing information, including the time from the start of the query recording to results display, the time required for the entire matching algorithm (trajectory generation, results ranking and display), and the time required by DTW.

If the participant made a mistake, for example forgot to mark a sign as one-handed, and needed to rerun the matching algorithm on one-handed signs, the entire time from their first attempt until they received appropriate results was logged. As real users of the system are expected to make mistakes, especially when learning the system, this provides more realistic usage timing data.

In Section 6.3.3, a comparison is provided of sign recognition results from the old and new systems on the same videos. To generate accuracy results on the old system, we imported the color videos recorded on the new system and used the same start and end frames as determined by the participant. As the old system is not intuitive to use, an individual experienced with the system performed all experiments. Since the old system offers the ability to incorporate handshape into the matching algorithm, we ran the signs twice, both with and without handshape, and recorded each sign's best rank between the two.

We also recorded informal timing data with the old system. Since the video had already been recorded on the new system, however, we did not include the recording time for these signs. Instead, we record the time from the beginning of sign video importation to the display of results. Timing data from the two systems is compared in Section 6.3.4.

6.3.3 Accuracy Results

System accuracy is computed as the percentage of signs whose correct match is found in the top k results returned by the system. Figures 6.5a–6.5e show system accuracy for the individual participants. It can be seen that in all but one case, the new system outperforms the old system to varying degrees. For example, for participant **P04**, the old system returned the correct sign in the top 10 matches for 20% of the query signs versus 66.7% with the new system.

We calculated an average accuracy for both systems, as well as and expected accuracy for a random system. Since there are fewer one-handed signs in the system

dictionary than two-handed signs, the maximum possible rank m is the number of two-handed signs, and the expected accuracy $f(r)$ at a rank level $r \in [1..m]$ is calculated:

$$f(r) = \begin{cases} \frac{2r}{N} & : r <= n \\ \\ \frac{r+n}{N} & : r > n \end{cases} \tag{6.1}$$

for number of one-handed signs n in a dictionary of size N, alternatively expressed:

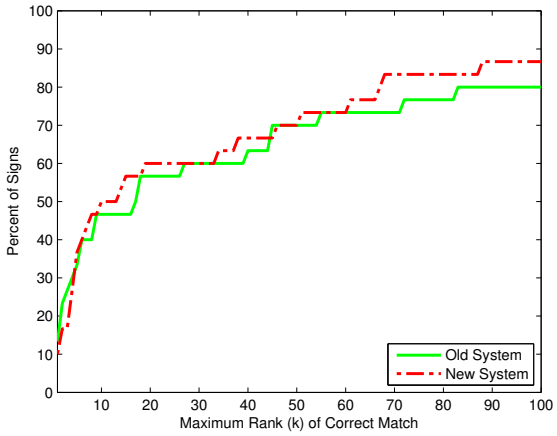
$$f(r) = \frac{r + \min(r, n)}{N}. \tag{6.2}$$

Figure 6.6a shows the accuracy for all rank levels and table 6.1 for a small subset. It can be seen that while both systems far outperform a system that randomly ranks the result signs, the new system shows a performance increase over the old. Figure 6.6b is a closer view of the accuracy in the maximum rank 1–100 levels. In the new system, for 62% of the query signs, the correct match is returned in the top 20 results, whereas this percentage drops to 46.7% in the old system. It is apparent that skeleton detection using scene depth information outperforms the skin color and motion-based hand tracking in the previous generation software. The same results can be seen in tabular format in table 6.1.

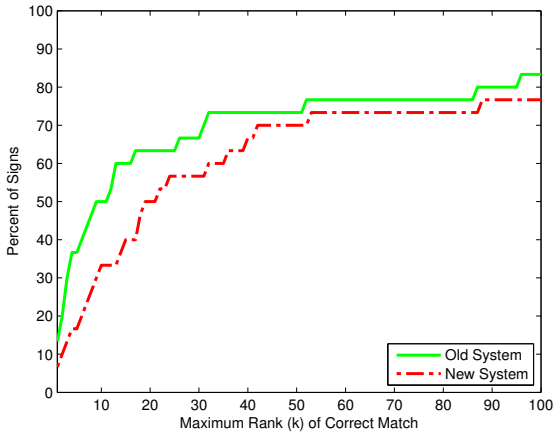
6.3.4 Query Time Results

The informal timing experiments show a significant performance increase in the new system. Table 6.2 shows the average and median times required by each user to perform a query, as well as the standard deviations. The *Average* row contains the averages of participants **P01**–**P05**, while the *System* row includes timing data from all participants in the calculations.

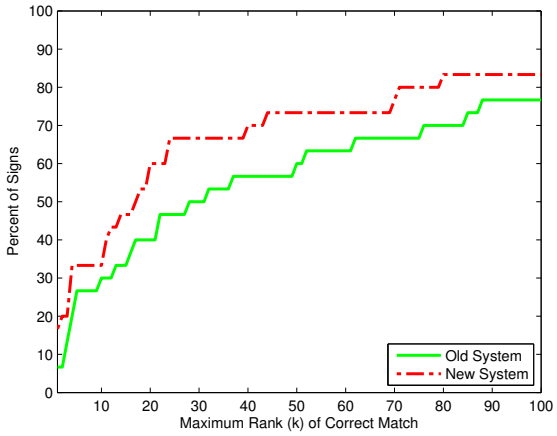
An experienced user performed the matching on the old system with the same videos. The timing data obtained here is informal, as it was obtained from a stop-



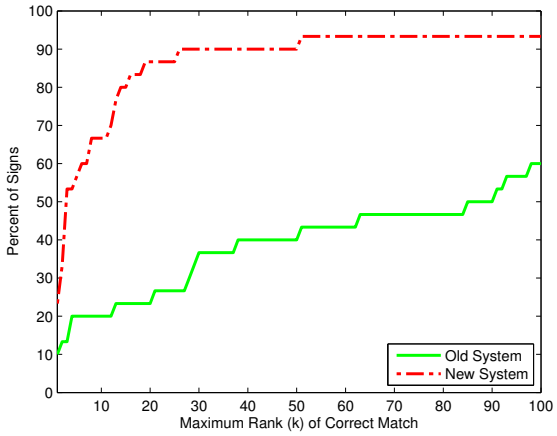
(a) Participant **P01**



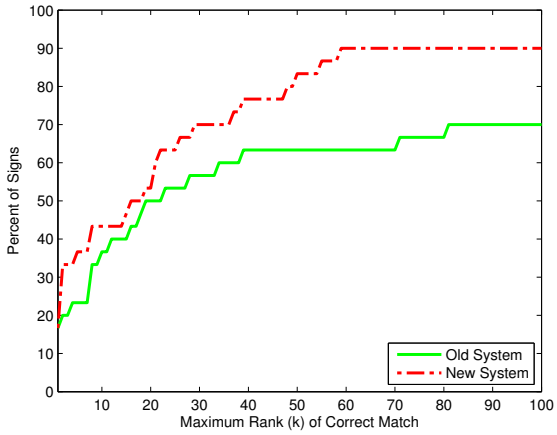
(b) Participant **P02**



(c) Participant **P03**

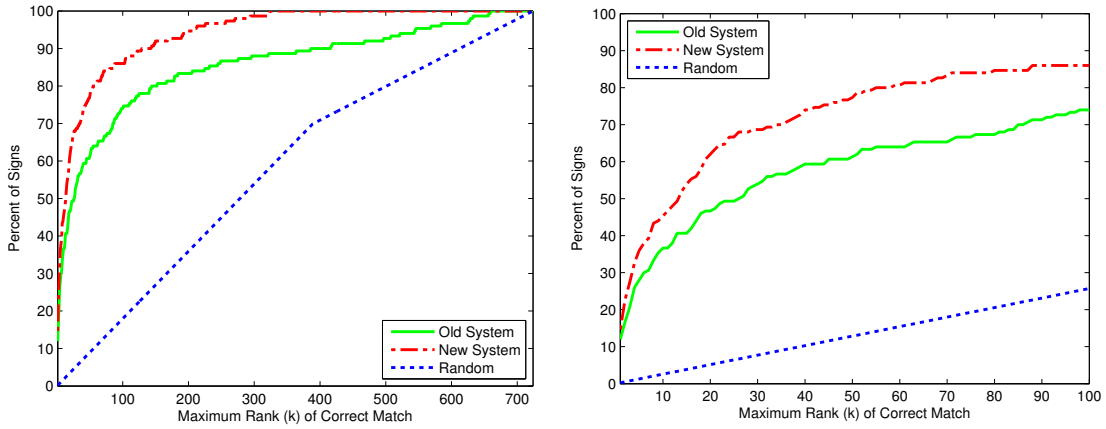


(d) Participant **P04**



(e) Participant **P05**

Figure 6.5: System sign recognition accuracy – green: old; red: new



(a) All ranks.

(b) Ranks 1–100.

Figure 6.6: Average System Accuracy Comparison

watch. Furthermore, it did not include the time to record the videos, since they were not recorded with this system. The timed portion consisted of importing the video into the system, marking the start and end frames as the participant marked them in the new system, initialization of the hand tracker, tracking, the matching algorithm, and results display. Once the first set of results was displayed, the timer was stopped. Whereas the new system showed an average query search time of 22.0 seconds, the old system had a mean time of 106.2 seconds per query. See table 6.3 for details.

Table 6.1: Accuracy of the Old and New Systems

Max Rank	Old System	New System
1	12.0%	14.7%
2	16.7%	22.7%
3	20.7%	27.3%
4	26.0%	32.7%
5	28.0%	36.0%
10	36.7%	45.3%
15	40.7%	54.0%
20	46.7%	62.0%
30	54.0%	68.7%
50	61.3%	77.3%

Table 6.2: Timing Data in Seconds for Study Participants.

Participant	Mean	Median	Std. Dev.
P01	13.1	11.0	6.18
P02	25.1	21.9	9.33
P03	15.1	14.8	3.19
P04	27.4	24.7	13.0
P05	29.2	27.5	9.93
Average	22.0	20.0	8.33
System	22.0	19.4	11.1

Table 6.3: Query Time Comparison

System	Mean	Median	Std. Dev.
Old	106.2	106.4	9.847
New	22.00	19.40	11.06

CHAPTER 7

LEVERAGING INTRA-CLASS VARIATIONS TO IMPROVE RECOGNITION

7.1 Introduction

It is clear that gesture and sign language recognition is a challenging research field in computer vision. Many popular probabilistic methods like Hidden Markov Models (HMM) [76] and Conditional Random Fields (CRF) [77] require large training sets to learn probability distributions. Neural network based approaches similarly require large training and validation sets to increase generalization and minimize overfitting to the training examples. This requirement often limits the vocabulary size of such systems. When a large vocabulary is desired, however, time and fiscal constraints may force researchers to restrict the size of the training set and thus limit the techniques that can be used for classification. As using only a few examples per gesture class precludes the use of many statistical and machine learning methods, researchers are often limited to exemplar-based recognition and similarity measures.

In such cases, Dynamic Time Warping (DTW) [58] is often used on hand position or other information to generate scores that serve as a measure of similarity to training examples [78, 79, 80, 52]. DTW is improved with the use of a well-designed feature vector that includes more than hand positions to represent the state of a gesture at each point in time [52]. The performance of DTW-based recognition, however, can suffer due to variations in gesture performance inherent in user-independent systems. The two methods presented in this chapter address this problem.

These methods build on past work in ASL recognition. The base method used as a measure of gesture similarity is DTW, a dynamic programming technique that

creates an optimal alignment of two sequences [58], in this case, the hand trajectories of test and model signs. It has the benefit of being able to warp the temporal dimension of the series, so that is can properly align gestures performed at different speeds. The score provided by DTW is a measurement of error in the alignment, so that a lower score indicates a better match.

Rather than solely using the hand positions to describe the trajectories, we modify the feature vector introduced in [52] and further adapted in [81] to use information available from RGB-D output and then use multiple passes through DTW to generate several scores per example sign. Each pass focuses on a different gesture property and size normalization technique. We also introduce a new set of features that describe the likelihood of the measured variations between the test and example sign in several geometric and positional properties. The scores from the features and from multiple DTW passes can then be linearly combined to improve recognition accuracy. See chapter 3 for reviews of the feature vector we use for our sign representation and DTW.

Both methods are based on the natural variations that occur when multiple persons perform the same gesture. The intuitive notion behind the methods is that individuals each have their own style and will perform the same gesture in different positions and orientations in the gesture space or at different sizes. One signer may perform a sign directly in front of their torso, while another may perform it slightly to the side. Some signers perform the sign with very minor hand movement, and still others exaggerate the performance and show large movements. If we can somewhat relax the assumptions of what these characteristics of each sign class should be like, we can potentially improve trajectory alignment. The goal, then, becomes to use knowledge of these inherent variations to improve gesture recognition accuracy. Two methods have been developed.

The first method models this intra-class variation (ICV) in the geometric and positional properties of same-class gestures to provide indications of likelihood that a test gesture belongs to the same class as a training gesture, given their observed differences in these properties. The idea is that each sign class may show variations in these properties in different degrees, so that the variations that are considered relatively normal in one sign may be unusual for another. These likelihoods can become features that, when linearly combined and used in conjunction with DTW scores, provide a better indication of gesture similarity.

The second method, Multiple-Pass DTW (MP-DTW), involves generating multiple DTW scores for a test gesture. The purpose of this method is to better account for the fact that individuals will locate a sign in a different area of the signing space than others and that they will alter the size of the trajectory to match their body type and signing style. Rather than generate likelihoods, as with ICVM, MP-DTW seeks to generate multiple similarity scores by effectively creating several classifiers that focus on different aspects of the sign and normalizing the size in different ways. The DTW technique on which this work is based, detailed in Chapter 3, expresses hand positions in a single coordinate system centered on the head and normalizes the gesture size based on size of the face. This works well when the testing and training subjects perform the gesture in roughly the same position and at the same size. In practice, however, there is wide variation in both where a gesture is performed and how large the space it occupies. It makes sense, then, to adjust both this coordinate system and how the gesture is sized, and to combine weighted scores from multiple DTW passes using the alternative centering and and resizing techniques.

We show that either of these contributions alone or in combination can provide a substantial improvement in accuracy over DTW, even when using noisy and unstable RGB-D skeletal data. We then compare these methods to HMM and LSTM network

approaches [82, 83, 84] to demonstrate the benefit of using them in gesture recognition systems that comprise a large vocabulary but have a small training set size.

7.2 Variation-Specific Related Work

Some methods, like HMMs and Parametric Hidden Markov Models (PHMMs) can explicitly model these variations [85, 86]. Wilson et al. examine PHMMs to handle variations in gestures that can provide emphasis or an indication of the degree to which a gesture applies, for example to indicate the size of an object [86], and compare them to standard HMMs. Depending on the size, the gesture may be exaggerated to a certain extent, and the PHMM method can model these variations. Their experiments, however are user-dependent and only test to see how well the methods can distinguish between two gestures. They also present a online-learning model adaptation method that, instead of explicitly modeling these variations, modifies the existing models to handle variations that may be considered noise and, thus, ignored. This variation is the variation in position and size of signs that does not particularly influence meaning but we are looking to handle with our proposed methods.

Ong et al. have several works in which they use spatial variations that alter sign meaning to gain additional information about the signs [87, 88, 89, 90, 91]. In our system, we do not attempt to recognize these modifiers, as they are infinite, but the techniques are perhaps applicable to our work.

These HMM-based methods are unusable in our case sue to the insufficient number of training examples to learn the transition probabilities and parameters. There are, however, some works that also approach the idea of specifically modeling the class variability in one form or another. Reyes et al. use inter-class and intra-class variability in joint positions to weight the contribution each joint has in DTW scoring for the classification of 5 action categories [92].

Some approach the issue by using additional personalization data from a new signer to tweak the recognition system for each user. Yin et al. use labeled examples to learn a distance metric and then adapt that metric to a new signer using unlabeled data from the new signer [93]. Their feature vectors are composed of 3D trajectory information in a face-centric coordinate system and handshape based on HOG features, and they use the learned and adapted metric in a K Nearest Neighbors (KNN) classifier. They do not seem to model any positional or size variations. To handle variations inherent in performances of same-class gestures by different subjects, Yao et al. generate a group of likelihood maximization-based classifiers and use the best one for each subject based on personalization data [94]. This is an idea that could be easily incorporated into our dictionary system, and an online learning algorithm could be used to retrain the system for each user.

Roussos et al. instead focus on modeling variation in handshapes to improve recognition accuracy and adapt the models to new signers [68]. Bautista et al. use intra-class variability in gesture feature vectors to learn a Gaussian Mixture Model and extend DTW to provide probabilistic scores rather than alignment error measurements [95]. Our method instead models variations in geometric properties of whole gesture trajectories to improve results and combines scores from multiple DTW classifiers focusing on different elements of the trajectories. It may be beneficial to employ multiple approaches.

7.3 Intra-Class Variation Modeling

This section introduces the intra-class variation modeling (ICVM) of several hand trajectory geometric properties and describes the new features that are generated from the models. These features give an indication of likelihood that a test

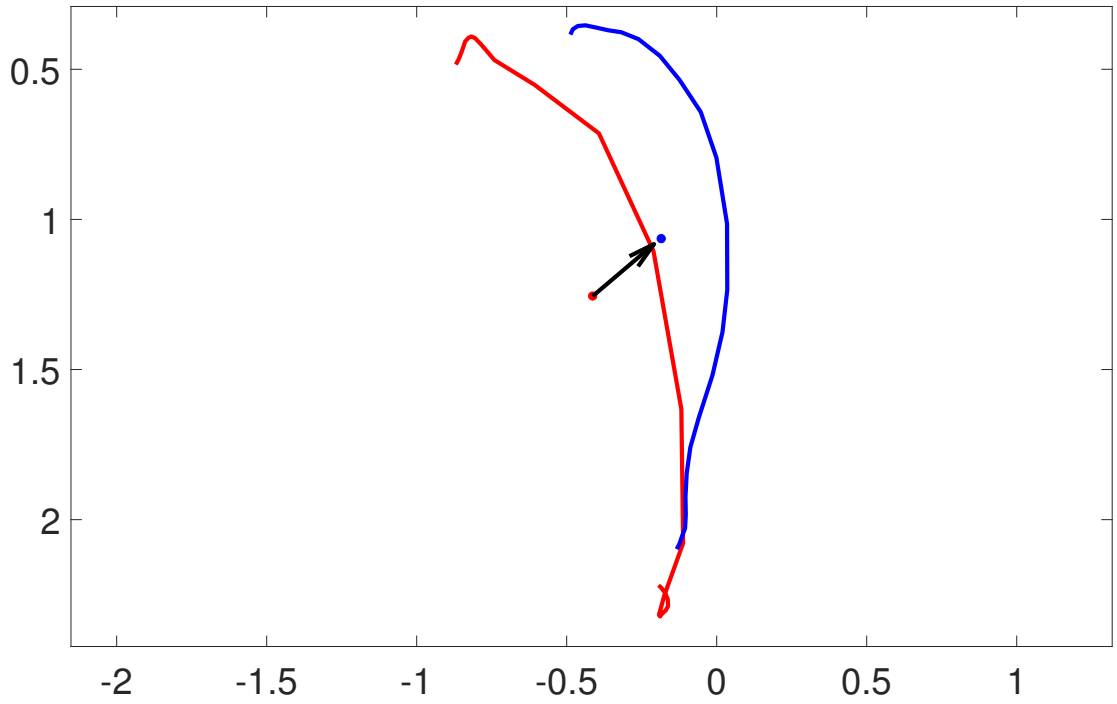


Figure 7.1: 2D property example. Shown are trajectories for two examples of the same sign. The measured variation is shown by the black arrow drawn from the centroid of one trajectory to the other.

sign would vary in these geometric aspects from a given model sign by the observed amount.

7.3.1 Method

Two sets—*LB1113* and *TB1113*—of one example each of 1,113 unique signs obtained from the ASLLVD [75] and a third set *GB1113* of the same signs, obtained from an alternate source, are used to train the variation models. *LB1113* and *TB1113* are each performed by a single signer, while *GB1113* consists of signs performed by multiple signers. Once the signs are expressed in the size-normalized, face-centric coordinate system described in section 3.2.2, we measure the difference in the properties between each sign of the same class. For example, figure 7.1 shows the dominant hand trajectory for two signs of the same class. The measured variation is between

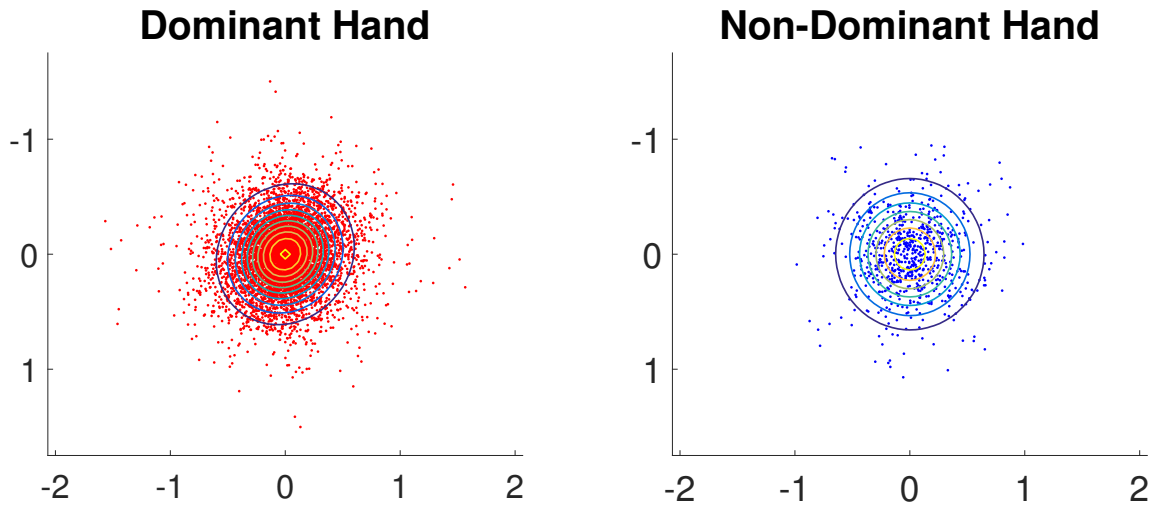


Figure 7.2: 2D Property Example: Intra-class variation plot for the centroid of the convex hull encompassing the dominant and non-dominant hand trajectories. The learned Gaussian models are overlaid.

the centroids of the convex hulls encompassing the two trajectories. The black arrow represents this difference vector.

Once the differences between all the same-class examples have been collected for the measured properties, a separate Gaussian is learned for each property to model the variation. Figure 7.2 plots the difference vectors as points for the convex hull centroid property of the dominant and non-dominant hand trajectories. It is clear that the single Gaussian that is overlaid is sufficient to model the variation in the property for each hand.

The variations in the following sign properties are modeled, separately for the dominant (d) and non-dominant (nd) hands in two-handed signs. The trajectory bounding box is defined as the box that extends from the leftmost to the rightmost hand position and from the topmost to the bottommost hand position throughout the sequence of video frames. Some properties, as indicated, are derived from the convex hull encompassing the set of hand positions throughout the sign.

1. γ_d and γ_{nd}: center of dominant and non-dominant hand trajectory bounding boxes, respectively.
2. ψ_d and ψ_{nd}: width of the dominant and non-dominant hand trajectory bounding boxes.
3. η_d and η_{nd}: height of the dominant and non-dominant hand trajectory bounding boxes.
4. α_d and α_{nd} : position of the dominant and non-dominant hand in the first frame of the sign.
5. ω_d and ω_{nd}: position of the dominant and non-dominant hand in the last frame of the sign.
6. λ_d and λ_{nd}: eigenvalue corresponding to the eigenvector describing the principle orientation of the dominant and non-dominant hand trajectory.
7. σ_d and σ_{nd}: smallest eigenvalue subtracted from the largest eigenvalue for the dominant and non-dominant hand: an indication of the strength in the trajectory orientation.
8. π_d and π_{nd}: perimeter of the dominant and non-dominant hand trajectory convex hulls.
9. ρ_d and ρ_{nd}: area of the dominant and non-dominant hand trajectory convex hulls.
10. ξ_d and ξ_{nd}: centroid of the dominant and non-dominant hand trajectory convex hulls.

Figure 7.3 plots the difference vectors of a set of test signs from the model signs as 2D points. The variation from same-class signs is plotted in yellow, while the variation from different-class signs is plotted in green. The Gaussian learned on the training set is overlaid on the figure, and it can be seen that it generalizes well to the test set.

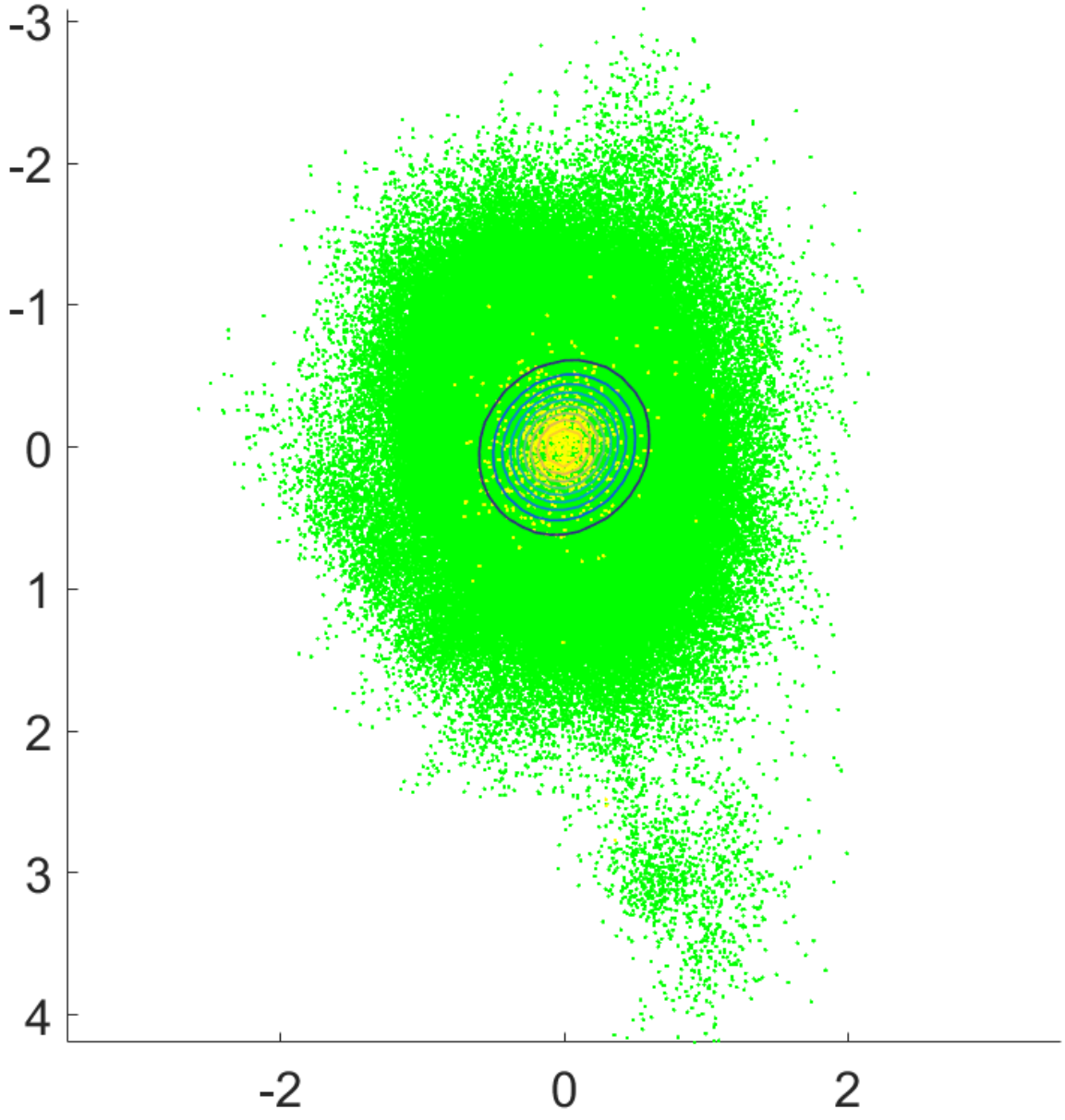


Figure 7.3: Plot of the measured differences as 2D points of all test signs from each example sign for the right hand trajectory convex hull centroid property. The differences from same-class signs are plotted in yellow and from dissimilar classes in green. The Gaussian model learned during training is overlaid to show that it generalizes well to test sets.

To generate a feature value $\phi_i(Q, M)$ for a given query sign Q using property i, the 1D difference x or 2D difference (x, y) is measured between Q and M for property i. The Gaussian for property i is evaluated using learned parameters (μ_i, σ_i) to calculate the feature value for 1D and 2D properties, respectively:

$$\phi_i(Q, M) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

$$\phi_i(Q, M) = \frac{1}{2\pi\sigma_{x_i}\sigma_{y_i}} e^{-\frac{(x-\mu_{x_i})^2}{2\sigma_{x_i}^2}+\frac{(y-\mu_{y_i})^2}{2\sigma_{y_i}^2}}$$

The feature values for multiple properties can be weighted and added to the base DTW scores to improve accuracy. As an example, if we were to take the convex hull centroid of a test sign and measure its difference vector, say $(2, -2)$, from a model sign, then the Gaussian evaluation would result in a low number, since that point would be located in the fringe of the distribution. This would indicate a low likelihood that the property of the test sign would vary from that of the example sign by that amount if they belonged to the same sign class. Consider, however, that the measured difference may be an outlier and the signs should, in fact, match. By examining other features, for example, trajectory width, convex hull perimeter, and strength of orientation, this likelihood would increase if the differences in those respects are minimal. This is effectively a relaxation of the assumptions of what characteristics a given sign class should have that may result in a correct match when otherwise the base method would not have done so.

The ICVM features and corresponding weights are trained using the *GB1113* dataset as the query set and the *LB1113* and *TB1113* datasets as the model sets. This dataset was chosen to train the features and weights due to the fact that it consists of multiple signers and is thus more likely to generalize and be more representative of

other individuals. A separate set of features and weights are learned for one-handed and two-handed signs as follows. Given the set of available feature properties

$$F = \{\gamma_d, \gamma_{nd}, \psi_d, \psi_{nd}, \eta_d, \eta_{nd}, \alpha_d, \alpha_{nd}, \omega_d, \omega_{nd},$$
$$\lambda_d, \lambda_{nd}, \sigma_d, \sigma_{nd}, \pi_d, \pi_{nd}, \rho_d, \rho_{nd}, \xi_d, \xi_{nd}\},$$

a subset of properties $N \subseteq F$ is simultaneously selected and weighted in a greedy manner: while an accuracy improvement above a threshold τ is achieved or there are remaining properties to be chosen, the property and weight combination (ν, β) that provides the best improvement when combined with previously selected features and base DTW scores is then included in N and removed from F. For one-handed signs, only the properties for the dominant hand are considered. We are interested in maximizing number of matches in the top 20 results returned by the system, since we feel the user would be willing to look through that many video examples to find the correct sign. For this reason, we define the above accuracy that is compared to the threshold τ to be the average accuracy at ranks 1–20. See Section 8.2 for a more detailed description of the measure of accuracy at various rank levels. In future work, it would make sense to weight the accuracies depending on rank, so that the lower ranks have greater importance in the calculations than do the upper ranks.

The *GB1113*-trained set of variation properties and weights are used in a our user-independent experiments described in chapter 8. We also trained a separate set of properies and weights for each test dataset using both manual and Kinect annotations for use in our user-dependent experiments, also in described in chapter 8. Though we also ran the experiments on each test set using the features and weights learned on the alternate signer test set using the manual and Kinect annotations, those learned on the *GB1113* set proved to generalize the best and are are used in all user-independent experiments. This is understandable, since the *GB1113* dataset

contains multiple signers and, as a result, encompasses more variation that reduces the chances of overfitting to a single signer.

The likelihood features can be used in various ways to increase recognition accuracy. Though a simple weighted linear combination thereof is used in this thesis as a proof of concept, other methods, such as SVMs and decision forest may prove more beneficial to filter out potential false positive matches or in a hierarchical classification scheme.

7.4 Multiple-Pass Dynamic Time Warping

Since different signers will likely vary the position and size of a gesture from one another, it does not make much sense to solely use the trajectory coordinate system and size normalization technique described in Section 3.2. It can be beneficial to use an alternate or even multiple alignment and size normalization methods for recognition. Figure 7.4 illustrates this benefit. The trajectories of two examples of the same sign class are shown. On the left, the trajectories are expressed in the face-centric coordinate system of the base method. On the right, the convex hull centroids become the origin of the coordinate system and a better alignment is achieved. If we combine this with a different resizing technique, for example using the height of the trajectory, it may further improve the trajectory alignment. The MP-DTW method leverages this opportunity by generating several DTW scores using alternative coordinate systems and size normalization.

7.4.1 Method

For the following two sets of gesture properties, d indicates the property applies to the dominant hand trajectory, nd to the non-dominant hand trajectory, and c to

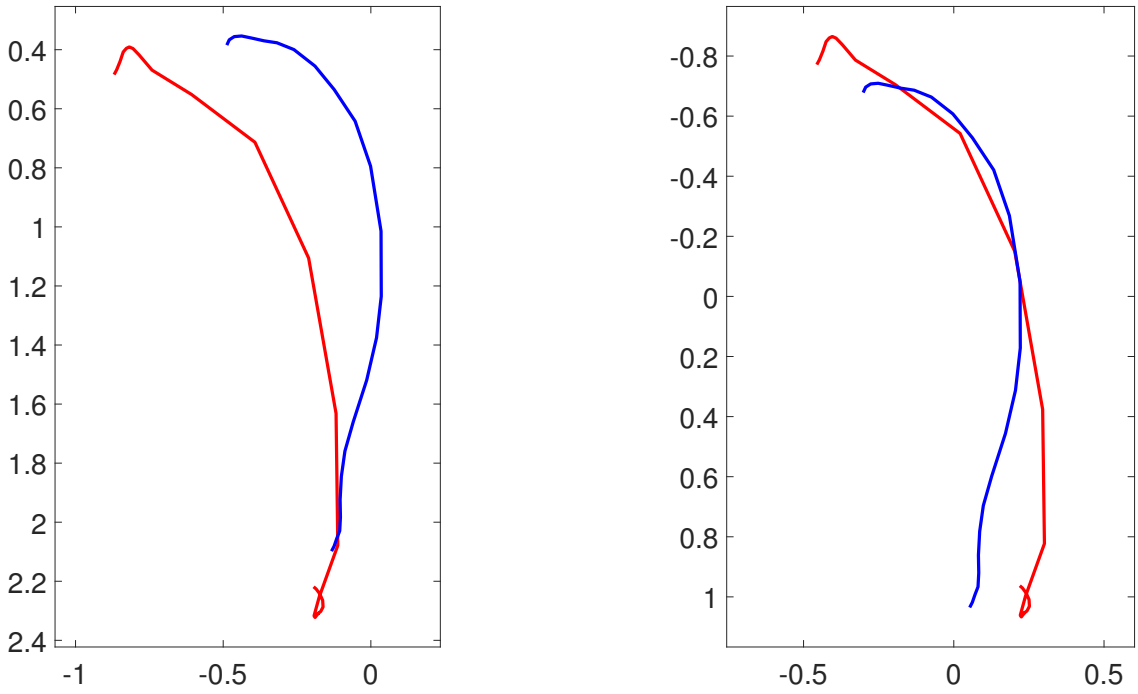


Figure 7.4: Motivation for MP-DTW. Left: gestures aligned on the face. Right: gestures aligned on the convex hull centroid. Using the centroid potentially gives a better DTW gesture alignment than the face. Combining multiple alignment methods results in better recognition accuracy.

the combined dominant and non-dominant hand trajectories. The set I of properties available for centering the gesture includes:

1. η: pixel coordinates of the head.
2. Γ_d, Γ_{nd}, Γ_c: center of dominant, non-dominant, and combined hand trajectory bounding boxes.
3. A_d and A_{nd} : position of the dominant and non-dominant hand in the first frame of the sign.
4. Ω_d and Ω_{nd}: position of the dominant and non-dominant hand in the last frame of the sign.
5. Ξ_d, Ξ_{nd}, Ξ_c: centroid of the dominant, non-dominant, and combined hand trajectory convex hulls.

6. M_d, M_{nd}, M_c: mean dominant, non-dominant, and combined hand position during gesture.

The set K of properties available to use for resizing the gestures includes:

1. Θ: face diagonal size
2. Ψ_d, Ψ_{nd}, Ψ_c: width of dominant, non-dominant, and combined hand trajectory bounding boxes.
3. H_d, H_{nd}, H_c: height of dominant, non-dominant, and combined hand trajectory bounding boxes.
4. Δ_d, Δ_{nd}, Δ_c: diagonal of dominant, non-dominant, and combined hand trajectory bounding boxes.
5. Λ_d, Λ_{nd}, Λ_c: diameter of the dominant, non-dominant, and combined hand trajectory, defined as as the largest distance between any two hand positions throughout the gesture.

To generate a feature value $\zeta_j(Q, M)$ for query Q and model M using centering and size-normalization property pair j, Q and M are centered and resized using j, and DTW is run to obtain a score. The score becomes the feature value that can be weighted and combined with other DTW passes.

$$\zeta_j(Q, M) = D_j(Q, M)$$

Using the *GB1113* dataset as the query set and *LB1113* and *TB1113* as model sets, a separate set of centering and size normalization properties are trained for one-handed and two-handed signs as follows. Given the set of centering properties

$$I = \{\eta, \Gamma_d, \Gamma_{nd}, \Gamma_c, A_d, A_{nd}, \Omega_d, \Omega_{nd}, \Xi_d, \Xi_{nd}, \Xi_c, M_d, M_{nd}, M_c\}$$

and the set of size normalization properties

$$K = \{\Theta, \Psi_d, \Psi_{nd}, \Psi_c, H_d, H_{nd}, H_c, \Delta_d, \Delta_{nd}, \Delta_c, \Lambda_d, \Lambda_{nd}, \Lambda_c\},$$

a subset of centering and resizing properties $\Upsilon = (\upsilon_1, ..., \upsilon_{|\Upsilon|})$, where each $\upsilon = (\iota_m, \kappa_n) \in I \times K$, is simultaneously selected (with replacement) and weighted in a greedy manner: while accuracy improvement is above a threshold τ, each combination of centering property ι_m and size normalization property κ_n is used to center and resize the sign for the DTW pass to obtain a score. The property combination and score weight (υ, β) that provide the best accuracy improvement when linearly combined with the base DTW score, previously selected MP-DTW feature scores, and ICVM features is included in Υ. For one-handed signs, only features for the dominant hand are considered. As with ICVM, the accuracy to compare to the threshold τ is defined as the average accuracy across ranks 1–20.

The *GB1113* trained set of property combinations and weights are used in a our user-independent experiments described in chapter 8. As with section 7.3, we also trained a separate set of properies and weights for each test dataset for use in our user-dependent experiments, also in described in chapter 8. The multiple signer *GB1113* set again proved to provide the best generalization to the test sets and were used for all user-independent MP-DTW experiments.

There were two ways to train the ICVM and MP-DTW features and weights. One was to train the ICVM features first and then select the best MP-DTW features that, when combined with the base DTW and ICVM scores, provide the best improvement. The other was to reverse the process and select MP-DTW weights first. While both provided comparable end results, training ICVM feature first seemed to provide the most stable results across the range of maximum ranks. See section 8.2 for our measure of accuracy.

7.5 Combining ICVM and MP-DTW

To generate a final score for an alignment between query sign Q and model sign M, the scores from DTW, MP-DTW, and ICVM features are linearly combined. Given base DTW score D_b, the set N of ICVM features, and the set Z of MP-DTW scores:

$$S(Q, M) = D_b(Q, M) + \sum_{i=1}^{|N|} \beta_i \phi_i(Q, M) + \sum_{j=1}^{|Z|} \beta_j \zeta_j(Q, M),$$

where $\phi_i \in N$ and $\zeta_j \in Z$. Though the lowest final score of the three examples for each sign class is used for ranking purposes in our experiments, it is possible to combine them into a single score. Since this version of DTW provides an error measurement of the alignment of two gestures, a lower score indicates a better match.

CHAPTER 8

EXPERIMENTS AND RESULTS

In this chapter, we demonstrate the significant improvement in accuracy that ICVM features and MP-DTW provide using both manual annotations and the noisy joint position data generated by Kinect skeleton detectors. Our results show that systems using a large vocabulary with few training examples per gesture class benefit from incorporating one or both of the techniques. It is clear that our method outperforms popular methods that rely on large training sets or smaller vocabularies.

8.1 Experimental Setup

To evaluate ICVM and MP-DTW, we performed a series of user-dependent and user-independent experiments using both manually annotated hand positions and Kinect joint positions for two datasets. The user-dependent experiments provide a measure of the full potential of the methods and illustrate their advantages in a system that learns from the user, as voice dictation software does. As one uses the system, a learning algorithm can adapt the Gaussian models and ICVM feature and MP-DTW score weights to fit that particular individual.

We used the *GB1113*, *LB1113*, *TB1113* datasets as models for the experiments. As described in sections 7.3.1 and 7.4.1, the *GB1113* dataset was used to select and learn features and weights for the various geometric properties and MP-DTW. As the sets consist of RGB video with manually annotated 2D hand positions, this set of experiments does not incorporate any 3D information from the Kinect. 3D trajectory matching and ICV modeling of 3D gesture properties remains for future work.

A separate score is calculated for each of the three examples per sign class, and the lowest score for each class is used for sign ranking purposes. To these baseline scores are added scores from the intra-class variation modeling and MP-DTW methods described in sections 7.3 and 7.4. Hand shape is not considered and is left for future work.

We used two datasets from [96] as test sets, both of which are of fluent signers and comprise a combination of 1-handed and 2-handed signs of varying complexity. The *JK850* dataset consists of 850 unique ASL signs, while *CK368* contains 368 unique signs and is a more difficult set due to the wide variation in size and the exaggerated performance of signs compared to the models. The combined accuracy across both datasets is reported.

We compare our method to HMMs and LSTM networks. The complexity of the models is relatively limited due to the lack of training examples. For user independent experiments, the model parameters are selected based on performance on the validation set. The results reported are against the final testing set of *JK850* and *CK368*.

For the LSTM network, a single layer is used before the LogSoftMax output layer, and the number of nodes is chosen based on validation performance. 190 and 320 nodes are used for one handed and two handed sign models, respectively. The network is trained with stochastic gradient descent and early stopping. Even with regularization techniques, the models were quick to overfit the data.

For HMMs, a separate model is trained for each class. For 1-handed signs, each model used only a single state. For 2-handed signs, each model contained 4 states. For both cases, the observation model is a single Gaussian. Final evaluation is performed using one-vs-all classification.

8.2 Measure of Accuracy

For these experiments, we define the measure of accuracy to be the percentage of signs whose correct match is ranked in the top k most similar signs for each $k \in \{1, 5, 10, 20, 30, 50, 100\}$. User-dependent experiment results are found in tables 8.1 and 8.2, while user-independent results are contained in tables 8.3 and 8.4. The best performance in each table is emphasized with a bold font. Both the *JK850* and *CK368* datasets are given equal representation in the results.

8.3 User Dependent Experiments

User-dependent sign recognition can be useful in a system that learns from the user over time, as does the Nuance Dragon NaturallySpeaking voice dictation software [97]. As one uses a system based on our proposed methods, a learning algorithm can adapt the Gaussian models and ICVM and MP-DTW score weights to fit that particular individual and improve performance. For these experiments, the ICVM and MP-DTW properties were selected and the weights trained using the test sets themselves, providing a measure of method potential and setting a goal for accuracy when training with different signers. Table 8.1 shows the results on the combined *JK850* and *CK368* datasets using manual annotations. We achieve a 9.2 percentage point increase in accuracy for rank $k \leq 10$.

Since it is not realistic to expect a gesture recognition system to have access to manually provided ground truth annotations of hand locations, we performed the same experiments using the hand positions from the Kinect skeleton detector. The results in table 8.2 show a significant increase in accuracy using these noisy data. As an example, recognition at rank $k \leq 10$ increases from 45.7% to 54.3% when using the automatic annotations.

Table 8.1: User-Dependent Accuracy: Manual Annotations

	Maximum Rank k of Correct Match						
	1	5	10	20	30	50	100
HMM	0.206	0.452	0.557	0.652	0.719	0.785	0.870
LSTM	0.165	0.415	0.552	0.668	0.730	0.808	0.890
DTW	0.293	0.558	0.685	0.793	0.839	0.889	0.947
DTW + ICVM	0.316	0.597	0.730	0.821	0.852	0.897	0.947
MP-DTW	0.317	0.628	0.755	0.844	0.874	0.913	0.951
MP-DTW + ICVM	**0.333**	**0.646**	**0.777**	**0.855**	**0.878**	**0.918**	**0.955**

Table 8.2: User-Dependent Accuracy: Kinect Annotations

	Maximum Rank k of Correct Match						
	1	5	10	20	30	50	100
HMM	0.100	0.259	0.326	0.401	0.462	0.551	0.684
LSTM	0.089	0.246	0.335	0.444	0.516	0.616	0.743
DTW	0.162	0.372	0.458	0.562	0.607	0.682	0.791
DTW + ICVM	0.173	0.408	0.530	0.602	0.646	0.715	0.816
MP-DTW	0.199	0.399	0.493	0.596	**0.669**	0.702	0.820
MP-DTW + ICVM	**0.204**	**0.431**	**0.543**	**0.622**	0.664	**0.730**	**0.827**

8.4 User-Independent Experiments

User independent experiments demonstrate the potential improvement these methods provide in a pre-trained gesture recognition system that does not learn from the user, as our system is currently configured. Table 8.3 shows the accuracy when using manual annotations. As can be seen, the best results again come from the combination of ICVM and MP-DTW. Table 8.4 provides results using the Kinect skeletal

Table 8.3: User-Independent Accuracy: Manual Annotations

	Maximum Rank k of Correct Match						
	1	5	10	20	30	50	100
HMM	0.158	0.359	0.459	0.582	0.647	0.725	0.831
LSTM	0.124	0.315	0.428	0.573	0.648	0.735	0.833
DTW	0.293	0.558	0.685	0.793	0.839	0.889	0.947
DTW + ICVM	0.298	0.590	0.714	0.802	0.849	0.897	0.950
MP-DTW	**0.336**	0.621	0.728	0.823	0.865	**0.908**	0.948
MP-DTW + ICVM	0.314	**0.625**	**0.731**	**0.824**	**0.867**	**0.908**	**0.957**

annotations. It shows an increase in accuracy from 45.8% to 49.5% for maximum rank $k = 10$.

Table 8.4: User-Independent Accuracy: Kinect Annotations

	Maximum Rank k of Correct Match						
	1	5	10	20	30	50	100
HMM	0.089	0.220	0.296	0.376	0.435	0.512	0.645
LSTM	0.060	0.201	0.282	0.397	0.476	0.572	0.701
DTW	0.162	0.372	0.458	0.562	0.607	0.682	0.791
DTW + ICVM	0.176	**0.394**	**0.498**	0.576	0.637	0.713	0.815
MP-DTW	0.192	0.385	0.492	0.578	0.632	0.701	0.809
MP-DTW + ICVM	**0.197**	0.392	0.495	**0.592**	**0.650**	**0.727**	**0.823**

It is clear from these results that the best overall improvement comes from combining MP-DTW and ICVM features and that the two methods far outperform the HMM and LSTM network approaches in large vocabulary systems with few training examples per gesture class. In future work, we will be experimenting with other ways

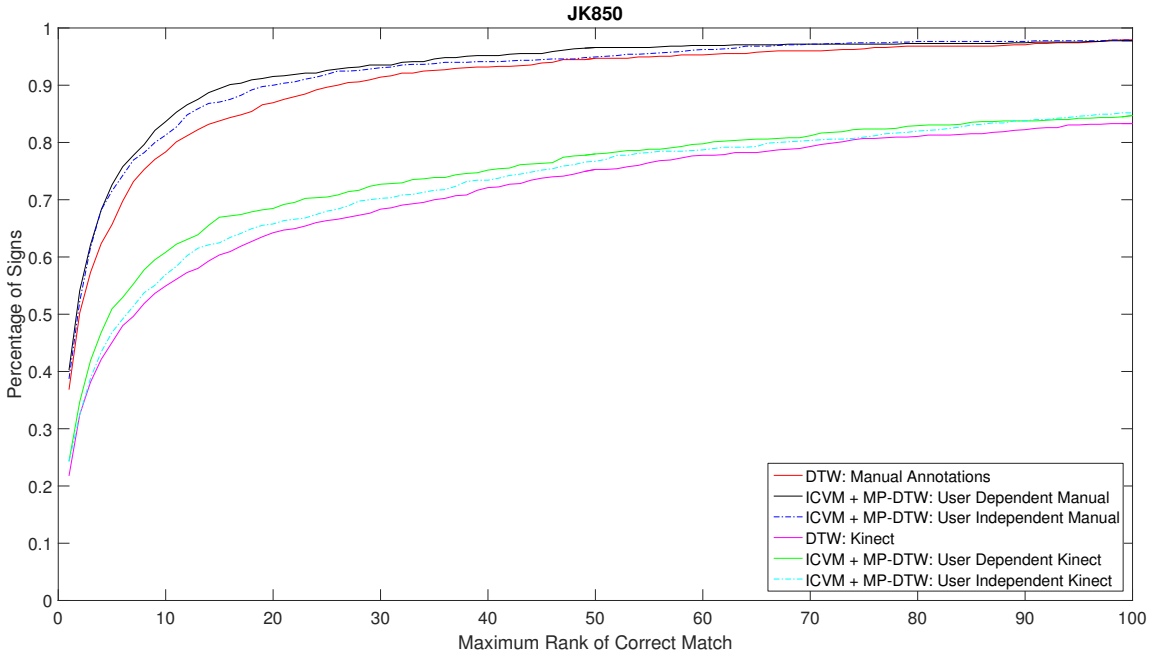


Figure 8.1: Accuracy plots for the *JK850* datasets. The plot shows the improvement using both Kinect and manual annotations in user-dependent and user-independent experiments.

to use the ICVM features, including random decision forests, SVMs, a cascade filtering of potential matches, and a hierarchical classification system. Secondly, due to time required to do so, we did not train the individual MP-DTW passes with their own set of DTW score component weights $\{s_1, ..., s_6\}$ as discussed in section 3 and instead left it for future work.

8.5 Further Analysis of Results

The best performance increases are on the *CK368* dataset, which contains the most variation in the location, size, and exaggeration level of the gestures compared to the models. Figures 8.1 and 8.2 plot the combined one-handed and two-handed accuracy on the two datasets using manual and Kinect annotations for both user-dependent and user-independent experiments. It is clear that the variation model-

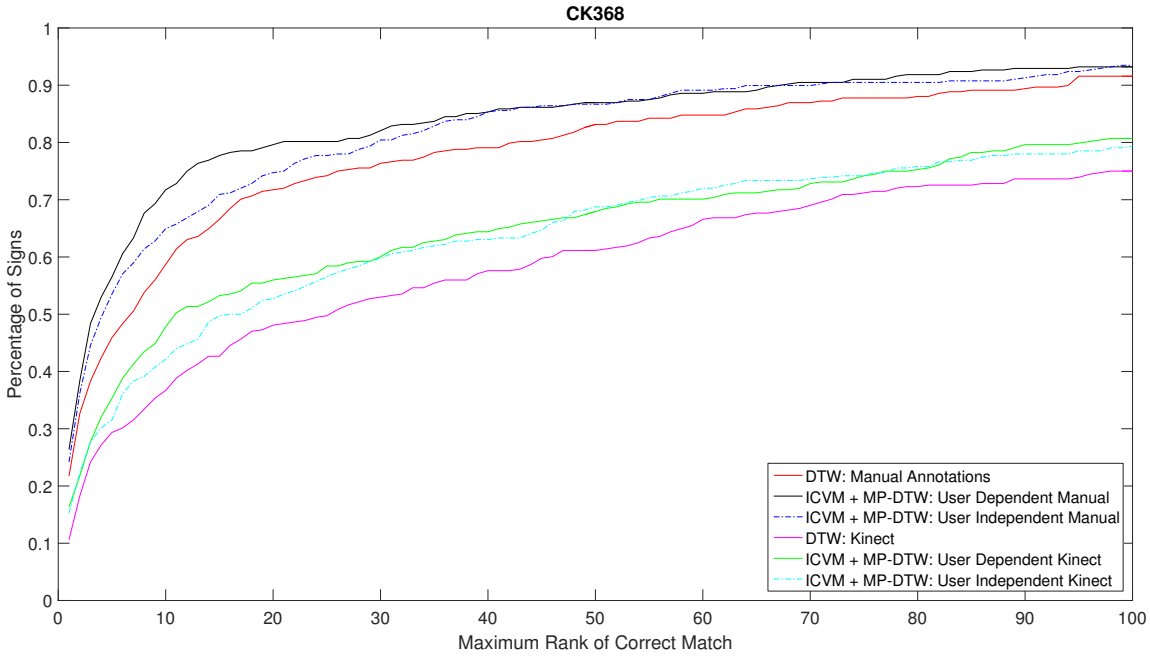


Figure 8.2: Accuracy plots for the *CK368* dataset. The plot shows the improvement using both Kinect and manual annotations in user-dependent and user-independent experiments.

ing and MP-DTW substantially improve accuracy, particularly on the more difficult *CK368* dataset. As the methods are designed to better handle wide variations in the performance of gestures, this make sense; the signer in *CK368* performed the signs with a greater degree of exaggeration, or emphasis, resulting in signs of different sizes and geometric properties of the hand trajectories.

One-handed signs are particularly difficult in sign language recognition, so it is beneficial to examine the performance separately on one-handed and two-handed signs. Figures 8.3 and 8.4 show the user-dependent and user-independent accuracy on the *CK368* dataset for both one-handed and two-handed signs using manual and Kinect annotations. Examining the plots shows that in most cases, the largest improvement in accuracy is on one-handed signs. One handed signs often have an arbitrary trajectory because the sign is static and the signer's inability to keep the

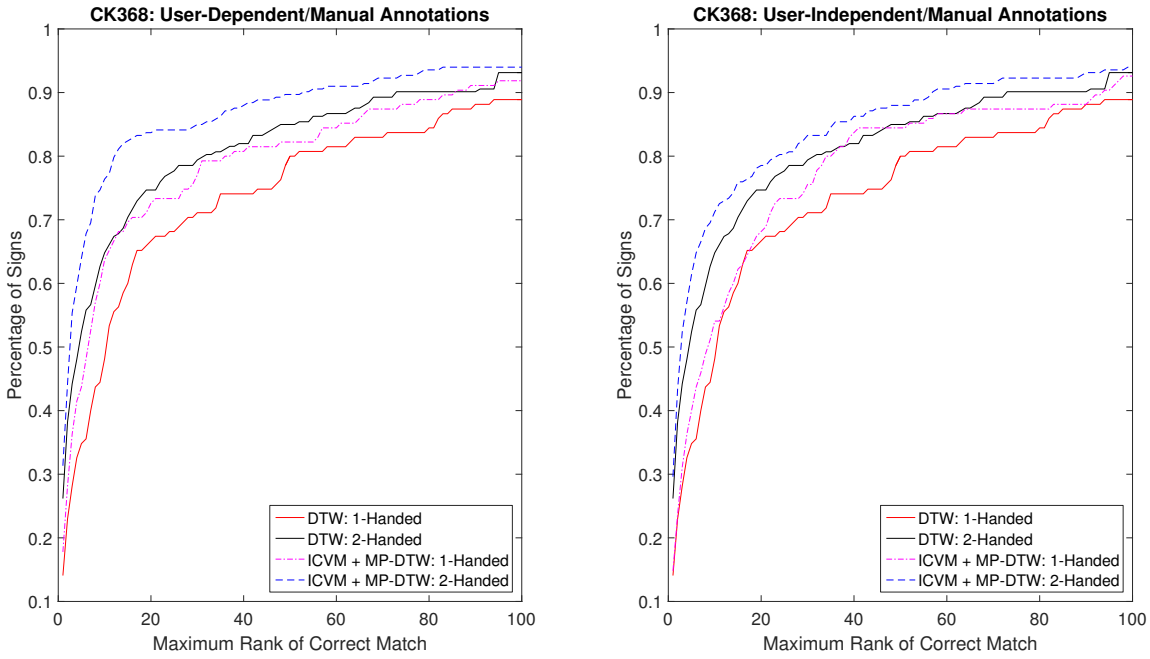


Figure 8.3: 1-Handed and 2-Handed user dependent and independent accuracy plots for the *CK368* dataset using manual annotations.

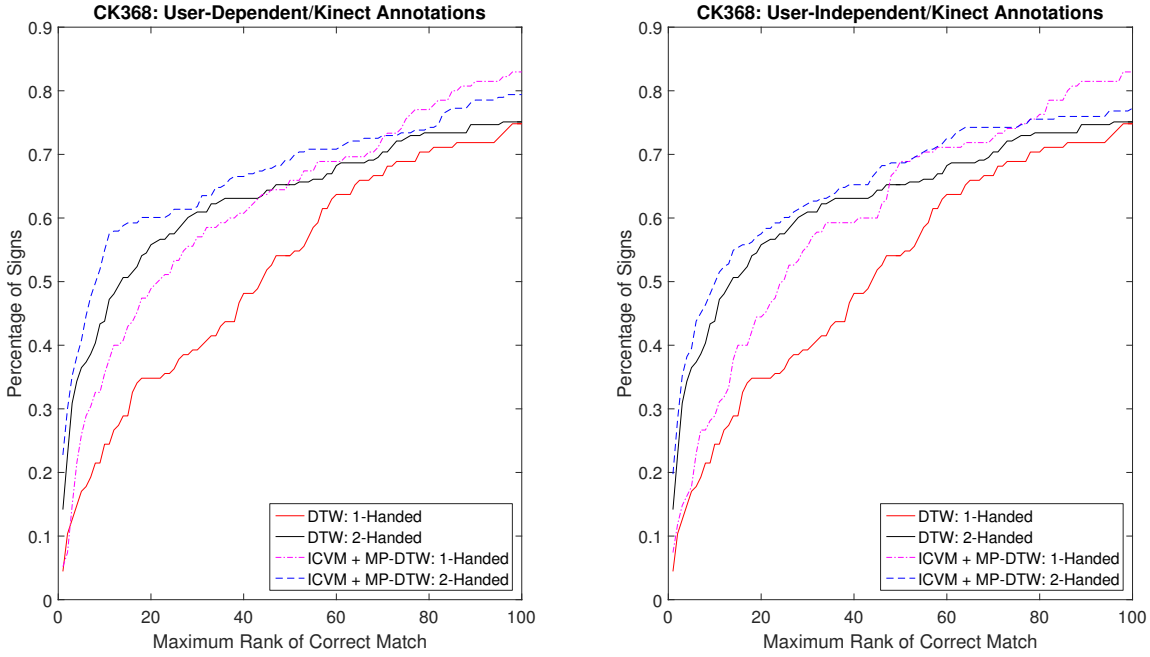


Figure 8.4: 1-Handed and 2-Handed user dependent and independent accuracy plots for the *CK368* dataset using Kinect annotations.

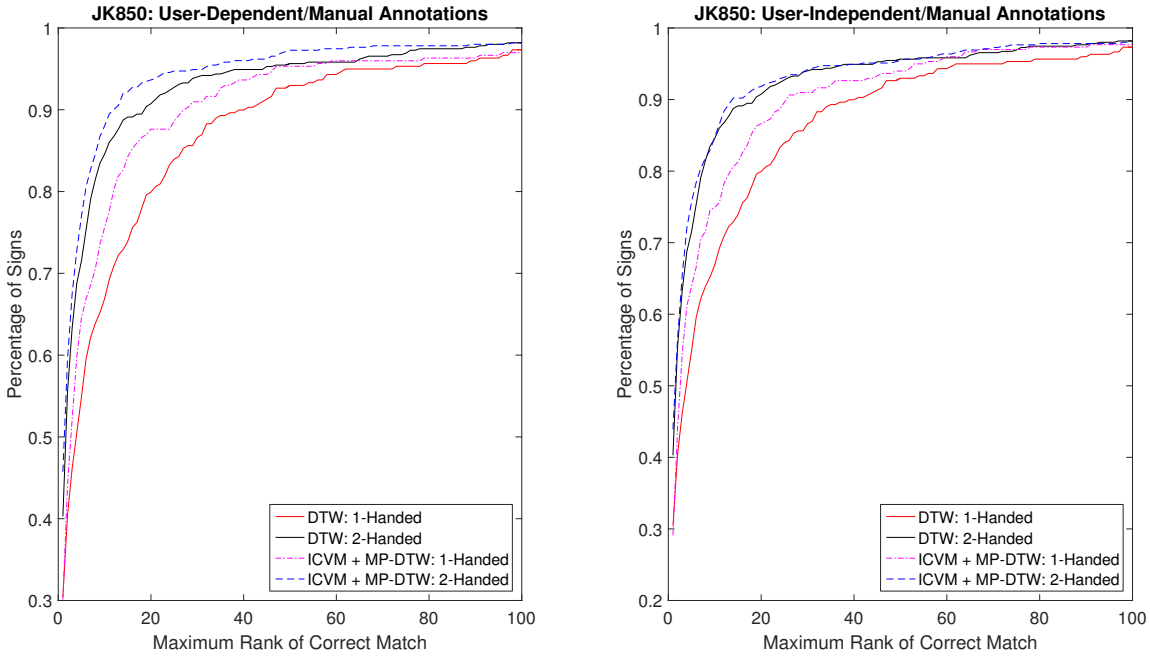


Figure 8.5: 1-Handed and 2-Handed user dependent and independent accuracy plots for the *JK850* dataset using manual annotations.

hand perfectly stationary causes random directions of movement. MP-DTW and ICVM somewhat relax the assumptions of what the trajectory should be like for a sign to belong to a particular class and thus result in a larger improvement on these signs.

Similarly, figures 8.5 and 8.6 show the user-dependent and user-independent accuracy on the *JK850* dataset for both one-handed and two-handed signs using manual and Kinect annotations.

8.6 Effect of Number of Features Used

This section presents a look at the effect of the number of ICVM features used on sign recognition accuracy. Figure 8.7 illustrated the improvement in accuracy that occurs as additional ICVM features are included in the similarity measure. It shows sign recognition accuracy, as measured by the method described in section 8.2, after

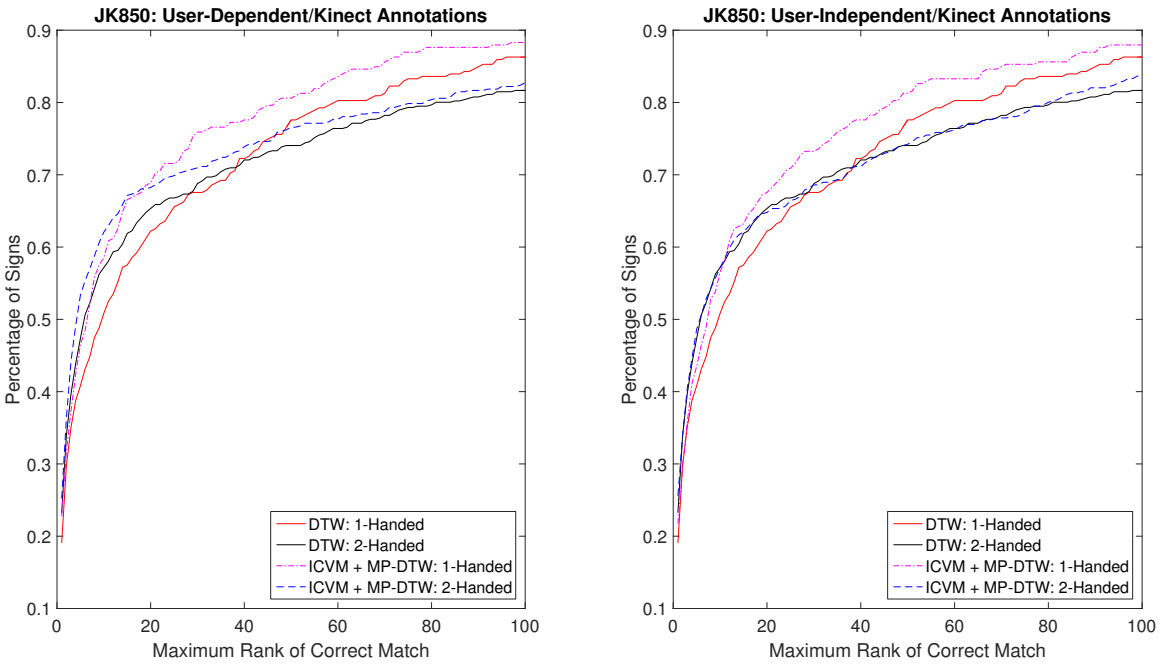


Figure 8.6: 1-Handed and 2-Handed user dependent and independent accuracy plots for the *JK850* dataset using Kinect annotations.

adding 1, 3, and 5 features. The best improvement in accuracy clearly comes from the addition of a single feature. The addition of a few more, however, especially when combined with MP-DTW (not shown in the graphic), can provide further improvement. In practice, the greedy feature chosing algorithm selected between 3 and 7 ICVM features and between 1 and 11 MP-DTW passes in addition to the base DTW pass.

8.7 Dictionary System User Experiments

To test how well ICVM and MP-DTW work on actual users of the dictionary system who are unfamiliar with ASL, we applied the methods to the signs used for the real-world experiments in Section 6.3.2. The experiments consisted of 5 participants without any knowledge of ASL, but we included an additional sixth signer here that was removed from the earlier experiments due to his authorship on the original paper

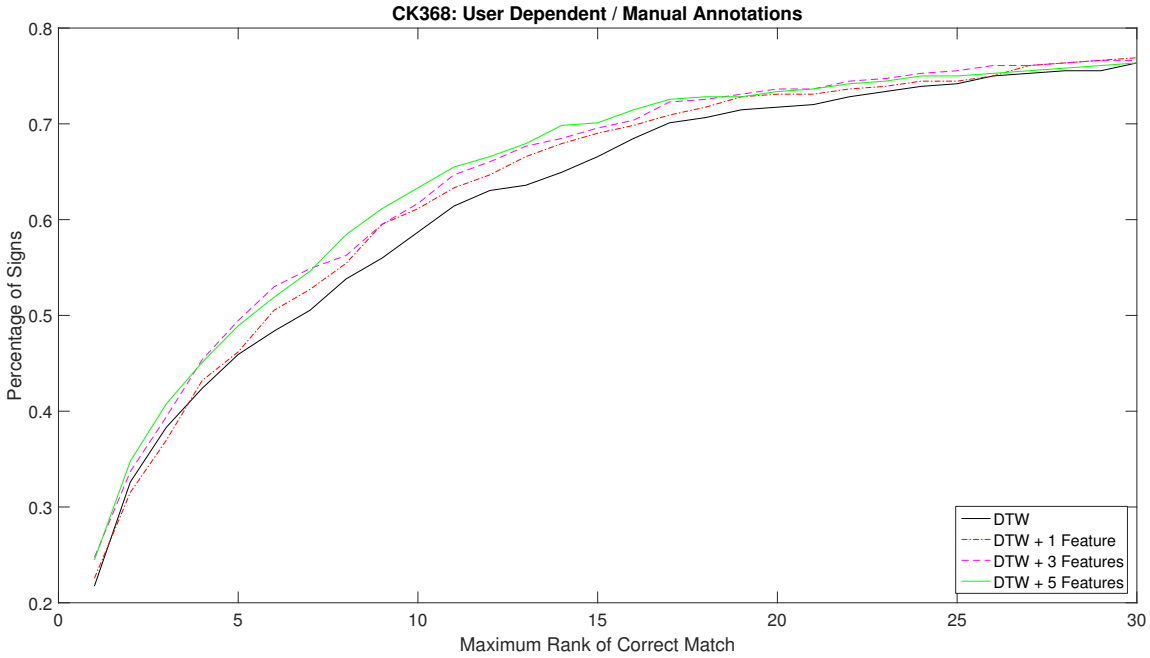


Figure 8.7: Shows the effect of the number of features used on accuracy for the *CK368* dataset using manual annotations and user-dependent experiments.

comprising that chapter. Each signer performed a set of 30 random signs from the system vocabulary. As an additional signer was included in this experiments, an updated baseline DTW accuracy is included in the result.

These experiments are perhaps most indicative of how well the methods would actually perform, since they involved participants who were not familiar with ASL or the signs they were performing. After being shown shown a video of a sign, they would perform it the best they could in front of the kinect and used the actual dictionary system to look up the meaning. Naturally, the users would show variations from the model signs.

Figure 8.8 shows the improvement in accuracy obtained by applying ICVM and MP-DTW to those signs. As with the other user-independent experiments in this section, the *GB1113* dataset was used to train the methods. It is clear from these

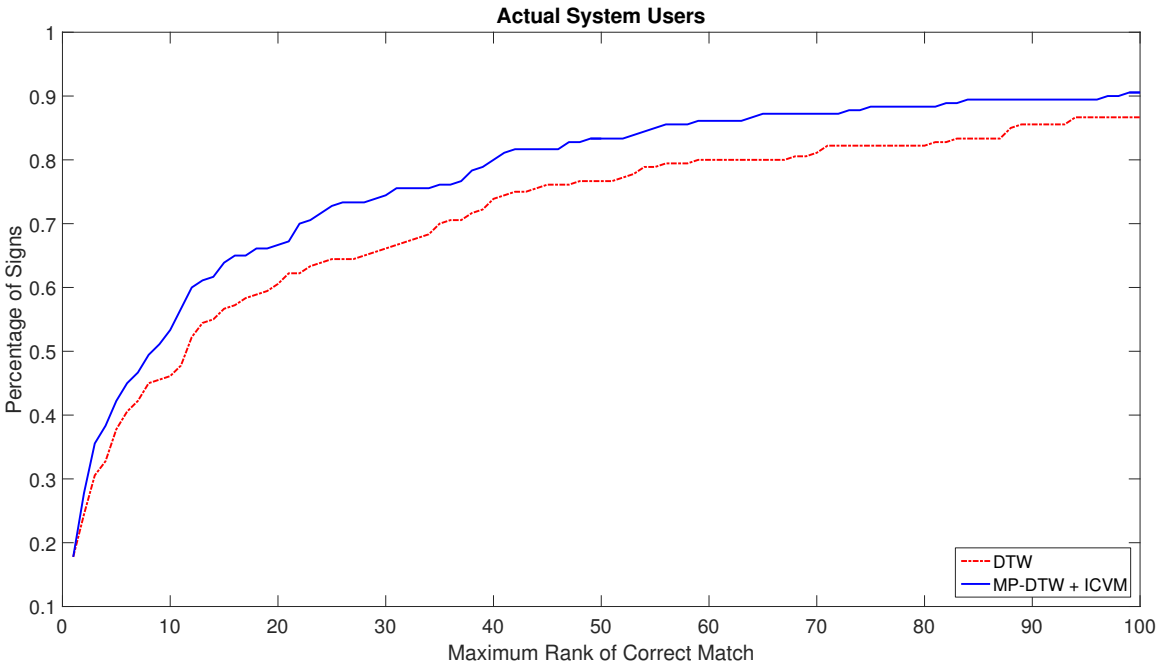


Figure 8.8: 1-Handed and 2-Handed user independent accuracy plots for actual users of the dictionary system.

results that the methods developed in this thesis show applicability in real-world systems and more work is warranted to further develop and expand the ideas.

8.8 Statistical Significance

To determine whether the achieved accuracy improvements have any statistical significance, we performed a series of paired sample t-tests on the classification results. For the tests, we separately calculate the significance at the maximum rank $k = 1$–$k = 30$ levels of accuracy to show the effectiveness across a range, as each user may be willing to browse a different number of videos to locate the desired sign. If we approach it from a binary classification standpoint and assign to each test sign a 1 if it was correctly matched in the top k results and a 0 if it was not matched in the top k results, we can compare the results using DTW and using MP-DTW with ICVM features.

Table 8.5: Paired Sample T-Tests

		User-Dependent		User-Independent	
		Manual	Kinect	Manual	Kinect
Rank	1	0.0363	0.0669	0.1745	0.1442
	2	0.0087	0.1216	0.1711	0.1891
	3	0.0006	0.0719	0.0072	0.2046
	4	0.0000	0.0303	0.0036	0.1201
	5	0.0000	0.0063	0.0013	0.0717
	6	0.0000	0.0029	0.0029	0.0375
	7	0.0000	0.0032	0.0071	0.0260
	8	0.0000	0.0060	0.0202	0.0856
	9	0.0000	0.0083	0.0397	0.0650
	10	0.0000	0.0090	0.0196	0.0325
	11	0.0003	0.0051	0.0549	0.0237
	12	0.0001	0.0102	0.0587	0.0310
	13	0.0001	0.0075	0.0329	0.0215
	14	0.0001	0.0096	0.0217	0.0312
	15	0.0002	0.0133	0.0133	0.0161
	16	0.0004	0.0190	0.0098	0.0261
	17	0.0005	0.0395	0.0139	0.0383
	18	0.0008	0.0650	0.0215	0.0459
	19	0.0074	0.0599	0.0389	0.0605
	20	0.0035	0.0859	0.0330	0.0914
	21	0.0067	0.0923	0.0226	0.0904
	22	0.0083	0.0710	0.0108	0.0371
	23	0.0094	0.0854	0.0231	0.0498
	24	0.0222	0.0996	0.0436	0.0491
	25	0.0246	0.0827	0.0730	0.0366
	26	0.0361	0.0749	0.1317	0.0241
	27	0.0392	0.0879	0.2254	0.0290
	28	0.0284	0.0780	0.1807	0.0357
	29	0.0267	0.0714	0.1593	0.0354
	30	0.0455	0.0927	0.1921	0.0285

Table 8.5 shows the results of the t-tests on user dependent and user independent results using the manual and Kinect annotations rounded to four decimal places. The statistically significant results at $\alpha = 0.05$ are emphasized in a blue font. The user-dependent experiments using manual annotations are the most significant with most rank levels showing significance at $\alpha = 0.01$ and even $\alpha = 0.001$. Also interesting is the significance in the user dependent experiments using Kinect annotations. Most of the accuracy improvement is in the lower ranks, so the significance being only in these lower ranks makes sense, and is indicative of the significance in a system that learns from the user as he or she interacts with the system. McNemar's tests provide similar results to the paired sample t-tests.

CHAPTER 9

DISCUSSION AND CONCLUSIONS

This thesis chapter summarizes the contributions made by this the work to the field of sign language search systems. Systems that have a large vocabulary but few training examples present unique challenges that were addressed in this document. There is also a large body of remaining work, and this chapter outlines the work that is currently being planned to both improve accuracy and ease of use of the ASL Video Dictionary System.

9.1 Contributions

This work has made contributions in a few key areas, primarily in the area of large vocabulary datasets, sign recognition methods and similarity measures using small training sets.

9.1.1 Datasets

We have presented a growing RGB-D body part detection and gesture recognition dataset that can be used in several fields of research and have developed some benchmarks for gesture recognition and hand detection. The multi-modal dataset allows researchers to run body part detection and gesture recognition experiments using multiple types of data, with access to fully annotated ground truth data. At present, two signers have been recorded, 1,113 signs from one and 750 signs from the other. A full set of annotations is included that consist of temporal segmentation

bounding boxes for various body parts and point locations for others, the occlusion status of joints, and 2D and 3D positions of the joints from the skeleton detector.

Also provided with the dataset is a set of benchmarks for body part detection and gesture recognition. As more signs are recorded and more annotations are completed, these benchmarks will be updated to provide the latest results and goals for developing new methods.

9.1.2 ASL Video Dictionary System

We have presented a new integrated RGB-D ASL video dictionary system that solves many of the problems associated with its former variant. The new system is fully contained in one software package, is faster, more intuitive, more automated, and more accurate.

9.1.3 Similarity Measure Improvements

This thesis presented two novel methods to improve the sign similarity measure and, thus, increase accuracy that are based on the natural variation in the way that different signers will perform gestures. They are effectively a relaxation of the assumptions of what each particular sign should look like, and we demonstrated their potential in a real-world system. We have demonstrated a significant improvement in accuracy over DTW alone in user dependent and user independent experiments using both manual and kinect annotations.

9.1.3.1 Intra-Class Variation Modeling

ICVM creates models for the variations found across signers in a set of geometric properties of sign hand trajectories. We can then measure the differences in these properties between a test sign and a model sign to generates a set of likelihood

features. The combination of these features gives an indication of the likelihood that the test sign would vary from the model sign by these amounts and can be used in various ways to filter false positives or, as is done in this thesis, as part of the sign similarity score itself. Using as few as a single feature as part of the score can provide a substantial improvement in accuracy.

9.1.3.2 Multiple-Pass Dynamic Time Warping

MP-DTW relaxes assumptions about where a sign should occur in the signing space and how much space it should occupy. It does so by creating multiple similarity scores from several DTW passes, each centering and resizing the the sign on a different set of features, rather than basing everything on facial coordinates and size. This helps account for one signer performing a sign more to the side and more exaggerated than the model signs. The multiple scores again can be used in various ways, but a simple linear combination proved effective. We demonstrated the benefit of using MP-DTW alone, but the best improvements in accuracy come from combining it with ICVM features.

9.2 Future Work

This section introduces some of the work that is being planned to continue research into automated sign recognition systems. It is broken into a few areas: 1) dataset improvements; 2) DTW and similarity measure improvements, including variation modeling; 3) sign dictionary system ease of use improvements.

9.2.1 Dataset

Work on the dataset is ongoing and improvements are currently being made. Development continues and the dataset is being expanded to contain a vocabulary

of 3000 signs. We have begun the process to record two additional fluent signers who will perform each sign both how the example signs are performed and how they as individuals prefer to sign them. As more signers are added to the set, machine learning and statistical methods will become available to researchers. Furthermore, as it is also a body part detection dataset, annotators are currently providing additional joint annotation data, such as shoulder and elbow positions. Finally, annotators are producing hand bounding boxes in the frames between signs in which the signer begins to move their hands from their sides into position to perform the sign and in the frames when the hands return to their sides. This will allow us to develop methods that track motion from the beginning of movement, through the sign, and to the end of movement.

9.2.2 DTW and Similarity Measure Improvements

This section presents areas of potential improvement in the DTW algorithm itself and in the similarity measure used, including variation modeling and MP-DTW. While ICVM and MP-DTW work well, a few changes in the way they are used can possibly provide substantial improvement.

9.2.2.1 ICVM

There is much that remains to be explored with the variation modeling. First, with the goal of improving accuracy using the Kinect skeleton annotations, we will extend the modeling into the third dimension and retrain the selected properties and corresponding weights as more signers are added to the dataset described in Chapter 4.

Second, the properties were all based on the face-centric and face-normalized coordinate system detailed in Chapter 3. As MP-DTW shows, the face isn't the

only possible origin of coordinate system we can use, and the face diagonal may not be the best size normalization property. We will fully explore the use of other property/resizing combinations for use with ICVM.

Third, we will also explore alternative ways to use the features. Their use in a linear combination with DTW score is a preliminary proof of concept that may not exploit the full potential of the models. One idea is to use the features in a probabilistic sense and modify DTW to generate probabilities instead of alignment error measurements, as is done in [95]. Another option is to use the set of features to filter out possible false positives. Through the use of random decision forests, SVMs, or a cascade of weak classifiers, we can examine the effect of various machine learning techniques.

9.2.2.2 MP-DTW

When we learned the property pairs to use for the origin of the coordinate system and for size normalizing the signs, we used the DTW score component weights $\{s_1..s_6\}$ that were trained in the face-centric and normalized system. If may be of benefit to learn a separate set of these weights for each property pair. That way, each MP-DTW pass is providing the best possible scores for maximum accuracy.

Secondly, there is a chance that MP-DTW relaxes the assumptions about where a sign can occur too much, so that two unique signs with similar trajectories, but different positions are confused with each other. It makes sense, then, to impose some kind of penalty measure for having to recenter the coordinate system, which the algorithm does not currently do. We will explore this fully in future work.

9.2.2.3 DTW Transition Costs

During frame alignment in DTW and its multiple-hand-candidate variant Dynamic Space-Time Warping (DSTW), we have the option of incorporating frame-to-frame transition costs into the scoring mechanism. The current dictionary system does not take such measures. Development of a transition cost can potentially improve recognition accuracy. Preliminary experimentation using the distance the hand traveled from frame to frame as a basis for the transition cost proved fruitless. This makes some sense, as this distance traveled is already encoded in the position information of the hands. Among the alternative options are creating a cost based on hand shape similarity from frame to frame as well as one based on hand detector scores. In future work, we will fully explore the options and make modifications to the feature vectors as needed to develop an acceptable transition cost.

9.2.2.4 3D Sign Trajectory Matching

Since a gesture or sign is not a planar event, we will explore bringing the trajectories into the 3rd dimension. This will require modification of the feature vector to include the z coordinates of the hand positions and motion directions. The signs will be aligned by setting the position of the head in the first frame to be the origin of the new coordinate system and will be resized in one of two ways: 1) so that the shoulder-to-shoulder distance is equal to 1 or 2) so that the distance from the signer's head to neck in the first frame is equal to 1.

The first set of experiments will make the assumption that the signer is in the same orientation in both the training and and testing videos and will perform no rotations of the point cloud, only translation and resizing. The second set of exper-

iments will explore whether rotating the point cloud so that the head and shoulders in the first frame are aligned on the XY plane improves recognition results.

9.2.3 Hand Tracking Improvements

As better hand detection equates to improved sign recognition results, we will investigate per-model hand detection methods. Two ideas may be explored. First is the idea of both 2D and 3D moving Gaussian heat maps to provide likelihoods for the position of the hands in each model sign class. These likelihoods can be used to narrow a list of hand candidates from a standard hand detector. In the training models, as the position of the hands moves throughout the duration of the sign, a Gaussian-shaped curve is moved through space. The longer the hand stays in a position, the more that region heats up and the higher the likelihood the hand will be found in that area in the test sign. As the hand moves away from a specific area, that area begins to cool, and the likelihood the hand will be found in that area drops. This will require the resizing of the query sign and its alignment to the model sign, accomplished by aligning the head positions. Figure 9.1 shows a heat map for the last frame of a sign that has a trajectory moving downward. Red represents a higher position likelihood; blue represents a lower position likelihood.

Another method that will be explored is to use model hand positions to generate a region of interest (ROI) in which to search for the hand in the first and last frames of the query sign; a tracker can then be used to track the hand throughout the sign. One option is to generate edge images of the handshape from the models and have a detector that looks for that handshape based on multi-scale search using Chamfer distance as a measure of similarity. A second option is to use a somewhat more sophisticated technique like dynamic affine-invariant shape-appearance model (Aff-SAM) [68]. To avoid forced matching, a score threshold can be empirically learned and

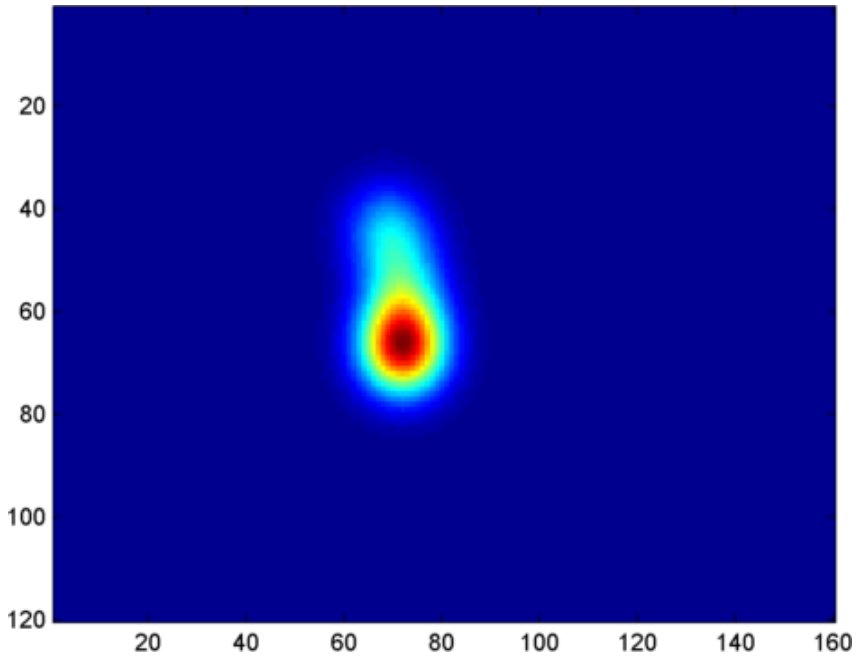


Figure 9.1: Example hand likelihood heat map

employed to indicate a successful or unsuccessful match. If there is an unsuccessful match, that sign class could be either discarded as a match or an alternate hand detection method used.

9.2.4 Dictionary System

We are currently planning improvements to the dictionary system, especially in the area of ease of use, which also have the potential to improve recognition rates.

9.2.4.1 Automatic 2-Handed Detection

One required step that should be eliminated to maximize system ease of use is the selection of sign handedness. Most of the longer dictionary system usage time resulted from the user failing to ensure the correct handedness was selected and having to rerun the match process after realizing what had happened. There are two ways to address this problem.

First, we can group the one-handed and two-handed models into a single set, and match signs against both types, instead of matching one-handed only to one-handed and two-handed to two-handed. Experimentation shows that this is a valid approach, and accuracy only suffers by a few percentage points. We could develop methods to improve this accuracy, but there might be a better approach.

The second way to address the problem is to use characteristics of two-handed signs to automatically determine sign handedness, thus relieving the user of the requirement and making the system easier to use. The two handed signs tend to share a few common characteristics. First, in one-handed signs, the non-dominant hand tends to stay out of the signing space, or the region in which signs are performed. See figure 9.2; the red rectangle demarcates the approximate signing space. This method can check for the presence of the non-dominant hand in the sign space. If it is present, further analysis is warranted to determine if it is part of the sign.

Furthermore, in two-handed signs, there is often interaction between the dominant hand and either the non-dominant hand or arm. The proposed method can also look for this interaction. It is often the case, however, that the dominant and non-dominant hand do not interact. In these cases, their movements tend to be either symmetric or anti-symmetric. See Section 9.2.4.2 for details and illustrations. The techniques discussed in that section can be applied here to test for handedness.

9.2.4.2 2-Handed Sign Sub-Classification

There are generally four types of two-handed signs commonly found in American Sign Language that will allow categorization into unique subclasses:

1. Symmetric: the two hands perform the same motion symmetrically.
2. Anti-Symmetric: the two hands perform a mirror image motion of each other.

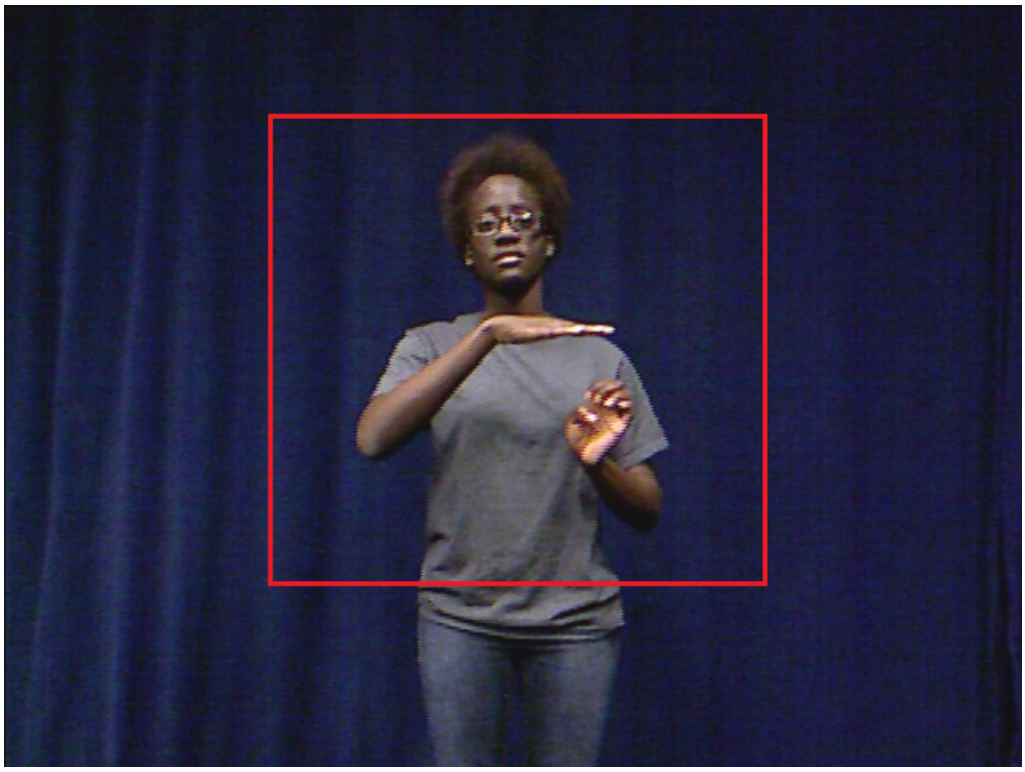
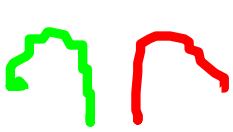


Figure 9.2: Signing space

3. Non-Dominant Static: the non-dominant hand is held stationary while the dominant hand is in motion.
4. Other: a combination of the above classes. Usually, the dominant hand performs some movement the non-dominant hand does not.

Figure 9.3 shows example trajectories for each two-handed sign type. The green lines represent the non-dominant hand trajectories, while the red lines correspond with the dominant hand trajectories. The hand movement characteristics of the above sign classes can be used to sub-classify the signs and narrow the search space in the database of known signs. We will develop a means to compare the motion of the hands and determine the appropriate subclass. The non-dominant static signs can be distinguished by the lack of significant motion of the non-dominant hand through the duration of the sign. It will be left to experimentation to define significant



(a) Anti-Symmetric Trajectory

(b) Symmetric Trajectory

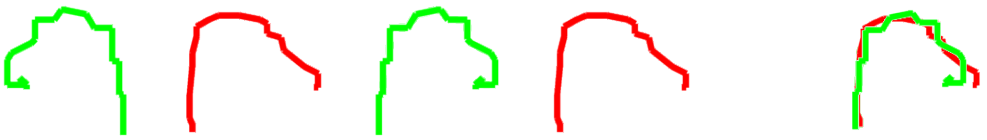
(c) Non-Dominant Static

(d) Other/Combination

Figure 9.3: Two-handed sign trajectory types.

motion. There are a few options that will be explored to check for symmetric and anti-symmetric signs.

First, the overall shapes of the dominant and non-dominant hand trajectories can be compared with a measure like Chamfer distance. This would require the alignment of the trajectories for symmetric signs (perhaps by aligning the starting points or by using Iterative Closest Points (ICP) to generate a best alignment), and in the case of anti-symmetric, the mirroring and alignment of trajectories. Figures 9.4 and 9.5 illustrate the process for anti-symmetric and symmetric two-handed signs,



(a) Original trajectories (b) Mirrored trajectory (c) Aligned trajectories

Figure 9.4: Anti-symmetric 2-handed trajectory comparison.

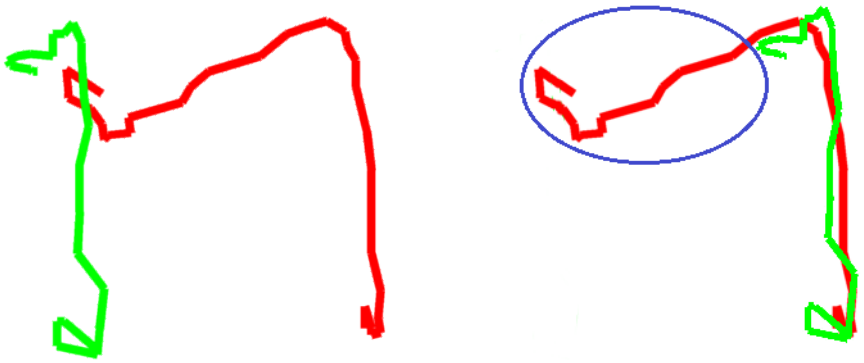
(a) Original trajectories (b) Aligned trajectories

Figure 9.5: Symmetric 2-handed trajectory comparison.

respectively. Whereas these illustrations show a comparison on a 2D image plane, the comparison can be brought into the third dimension by aligning the trajectories in space.

The above proposed methods, however, ignore the temporal aspect of a sign. Another method we will explore to compensate for this is to use DTW to determine how well the two hand trajectories align in both time and space. The trajectories will be expressed in a new coordinate system with their starting position in the first frame at the origin, and the non-dominant trajectory will need to be mirrored for the anti-symmetric comparison.

It remains to be determined how the mixed case will be tested. One option is to employ a modification to DTW, called gesture spotting, to look for the non-dominant trajectory inside the dominant trajectory (both mirrored and not mirrored). If there



(a) Original trajectories (b) Spotted trajectory

Figure 9.6: Spotting non-dominant hand trajectory

is a large portion (to be defined by experimentation) of the dominant trajectory that is not matched, this could be taken as an indication of the mixed status. Figure 9.6 shows an illustration of the idea. The blue circle marks the unmatched portion of the dominant hand trajectory.

9.2.4.3 Hand Shape

A large amount of the information conveyed in a sign is found in the shape and configuration of the hands. One of the top goals to improve the dictionary system is to incorporate automatic hand shape generation and comparison. By comparing the hand shapes in the first and last frames of a test and model sign, the accuracy can potentially be improved. To do so, the system must be able to expand from a single point for the hand as generated by the skeleton detection algorithm to a set of points corresponding to the entire hand. We must be able to cluster the pixels belonging to the hand and not the surrounding context. Furthermore, we will need a rotation invariant representation of the hand configuration.

9.3 Conclusions

In this thesis we have presented a new dataset for gesture recognition and body part detection research. In its final form, the dataset will contain a larger vocabulary than other publicly available dataset and will have a more complete set of annotations. From this dataset, we have generated benchmarks for gesture recognition and hand detection to serve as goals for the improvement of developed methods. We have also presented two novel improvements to exemplar-based large vocabulary gesture recognition with few training examples. The first, ICVM, generates a set of likelihoods that a test sign belongs to the same class as an example. The second, MP-DTW, helps account for variations in the position and scale of same-class signs in user-independent recognition systems. We have demonstrated a significant improvement in accuracy using the two methods. Furthermore, we have introduced a new fully-integrated ASL video dictionary system that is faster, more accurate, and easier to use than past variants. Finally, we have laid out several tracts of research to provide direction for future work in gesture recognition and body part detection.

REFERENCES

[1] J. Schein, *At home among strangers.* Washington, DC: Gallaudet U. Press, 1989.

[2] H. Lane, R. J. Hoffmeister, and B. Bahan, *A Journey into the Deaf-World.* San Diego, CA: DawnSign Press, 1996.

[3] R. Tennant, *American Sign Language handshape dictionary.* Washington, D.C: Gallaudet University Press, 2010.

[4] Learn sign language ASL dictionary lessons. [Online]. Available: http://www.handspeak.com/

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[6] Y. Wang, D. Tran, Z. Liao, and D. Forsyth, "Discriminative hierarchical part-based models for human parsing and action recognition," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3075–3102, Oct. 2012.

[7] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 2642–2649.

[8] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff, "Action recognition and localization by hierarchical space-time segments," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 2744–2751. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2013.341

[9] Y. Song, L. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 3562–3569. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2013.457

[10] C. Lu, J. Jia, and C. Tang, "Range-sample depth feature for action recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 772–779. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.104

[11] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 588–595. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.82

[12] Y. Song, L. P. Morency, and R. Davis, "Action recognition by hierarchical sequence summarization," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3562–3569.

[13] B. Fernando, E. Gavves, J. O. M., A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 5378–5387.

[14] O. Alsharif, T. Ouyang, F. Beaufays, S. Zhai, T. Breuel, and J. Schalkwyk, "Long short term memory neural network for keyboard gesture decoding," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 2076–2080.

[15] B. Bauer, H. Hienz, and K.-F. Kraiss, "Video-based continuous sign language recognition using statistical methods," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2. IEEE, 2000, pp. 463–466.

[16] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney, "Modeling image variability in appearance-based gesture recognition," in *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, 2006, pp. 7–18.

[17] C. Vogler and D. Metaxas, "Parallel hidden markov models for american sign language recognition," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 116–122.

[18] M.-H. Yang and N. Ahuja, "Recognizing hand gestures using motion trajectories," in *Face Detection and Gesture Recognition for Human-Computer Interaction*. Springer, 2001, pp. 53–81.

[19] Y. Cui and J. Weng, "Appearance-based hand sign recognition from intensity image sequences," *Computer Vision and Image Understanding*, vol. 78, no. 2, pp. 157–176, 2000.

[20] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments - PETRA '11*. New York, New York, USA: ACM Press, 2011, p. 1. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2141622.2141647

[21] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. Escalante, "Chalearn gesture challenge: Design and first results," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 1–6.

[22] J. Wan, Q. Ruan, W. Li, and S. Deng, "One-shot learning gesture recognition from RGB-D data using bag of features," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2549–2582, 2013.

[23] J. Konecný and M. Hagara, "One-shot-learning gesture recognition using HOG-HOF features," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2513–2532, 2014.

[24] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, "Multi-layered gesture recognition with kinect," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 227–254, Jan. 2015.

[25] S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: real-time action recognition," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2617–2640, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2567745

[26] M. Cuturi, J. Vert, Ø. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*, 2007, pp. 413–416. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2007.366260

[27] J. C. T. Pfister and A. Zisserman, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*. Cham: Springer International Publishing, 2014, ch. Domain-Adaptive Discriminative One-Shot Learning of Gestures, pp. 814–829.

[28] V. Pitsikalis, A. Katsamanis, S. Theodorakis, and P. Maragos, "Multimodal gesture recognition via multiple hypotheses rescoring," *Journal of Machine Learning Research*, vol. 16, pp. 255–284, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2789281

[29] T. Kim, G. Shakhnarovich, and K. Livescu, "Fingerspelling recognition with semi-markov conditional random fields," in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, 2013, pp. 1521–1528.

[30] R. Yang, S. Sarkar, and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 462–477, March 2010.

[31] R. Yang, S. Sarkar, and B. L. Loeding, "Enhanced level building algorithm for the movement epenthesis problem in sign language recognition," in *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2007.383347

[32] S. Sarkar, B. L. Loeding, R. Yang, S. Nayak, and A. Parashar, "Segmentation-robust representations, matching, and modeling for sign language," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2011, Colorado Springs, CO, USA, 20-25 June, 2011*, 2011, pp. 13–19. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2011.5981695

[33] D. Kelly, J. M. Donald, and C. Markham, "Continuous recognition of motion based gestures in sign language," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, Sept 2009, pp. 1073–1080.

[34] S. Nayak, S. Sarkar, and B. Loeding, "Unsupervised modeling of signs embedded in continuous sentences," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, June 2005, pp. 81–81.

[35] ——, "Automated extraction of signs from continuous sign language sentences using iterated conditional modes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 2583–2590.

[36] S. Nayak, K. Duncan, S. Sarkar, and B. L. Loeding, "Finding recurrent patterns from continuous sign language sentences for automated extraction of signs," *Journal of Machine Learning Research*, vol. 13, pp. 2589–2615, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2503325

[37] D. Kelly, J. M. Donald, and C. Markham, "Weakly supervised training of a sign language recognition system using multiple instance learning density matrices," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 526–541, April 2011.

[38] G. Yao, H. Yao, X. Liu, and F. Jiang, "Real time large vocabulary continuous sign language recognition based on op/viterbi algorithm," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 312–315.

[39] R.-H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on.* IEEE, 1998, pp. 558–567.

[40] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 63:1–63:8, 2009.

[41] I. N. Sandjaja and N. Marcos, "Sign Language Number Recognition," in *2009 Fifth International Joint Conference on INC, IMS and IDC.* Ieee, 2009, pp. 1503–1508. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5331335

[42] H. Cooper, E. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," *Journal of Machine Learning Research*, vol. 13, pp. 2205–2231, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2503313

[43] H. Wang, X. Chai, Y. Zhou, and X. Chen, "Fast sign language recognition benefited from low rank approximation," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 1, May 2015, pp. 1–6.

[44] Kinect Sign Language Working Group, "DEVISIGN Database," 2016. [Online]. Available: http://vipl.ict.ac.cn/homepage/KSL/data.html

[45] J. Han, G. Awad, and A. Sutherland, "Modelling and segmenting subunits for sign language recognition based on hand motion analysis," *Pattern Recognition Letters*, vol. 30, no. 6, pp. 623–633, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2008.12.010

[46] G. Awad, J. Han, and A. Sutherland, "Novel boosting framework for subunit-based sign language recognition," in *Proceedings of the International Conference on Image Processing, ICIP 2009, 7-10 November 2009, Cairo, Egypt*, 2009, pp. 2729–2732. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2009.5414159

[47] J. Han, G. Awad, and A. Sutherland, "Boosted subunits: a framework for recognising sign language from videos," *IET Image Processing*, vol. 7, no. 1, pp. 70–80, 2013. [Online]. Available: http://dx.doi.org/10.1049/iet-ipr.2012.0273

[48] J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, Nov 2008.

[49] J. Zieren and K.-F. Kraiss, "Robust person-independent visual sign language recognition," in *Proceedings of the Second Iberian Conference on Pattern Recog-*

nition and Image Analysis - Volume Part I, ser. IbPRIA'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 520–528.

[50] T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman, "Minimal training, large lexicon, unconstrained sign language recognition," in *In Proceedings of the 15th British Machine Vision Conference*, 2004.

[51] 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 *Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT)*, 2010, pp. 11–14.

[52] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar, "A system for large vocabulary sign search," in *Proceedings of the 11th European conference on Trends and Topics in Computer Vision - Volume Part I*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 342–353. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35749-7_27

[53] A. Stefan, H. Wang, and V. Athitsos, "Towards automated large vocabulary gesture search," *Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments - PETRA '09*, pp. 1–8, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1579114.1579130

[54] H. Wang, A. Stefan, and V. Athitsos, "A similarity measure for vision-based sign recognition," in *Universal Access in Human-Computer Interaction. Applications and Services*, ser. Lecture Notes in Computer Science, C. Stephanidis, Ed. Springer Berlin Heidelberg, 2009, vol. 5616, pp. 607–616. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02713-0-64

[55] A. Agarwal and M. Thakur, "Sign language recognition using microsoft kinect," in *Contemporary Computing (IC3), 2013 Sixth International Conference on*, Aug 2013, pp. 181–185.

[56] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the kinect," in *Proceedings of the 13th International Conference on Multimodal Interfaces*, ser. ICMI '11. New York, NY, USA: ACM, 2011, pp. 279–286. [Online]. Available: http://doi.acm.org/10.1145/2070481.2070532

[57] F. Pedersoli, S. Benini, N. Adami, and R. Leonardi, "Xkin: an open source framework for hand pose and gesture recognition using kinect," *The Visual Computer*, pp. 1–16, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00371-014-0921-x

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

[59] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, and A. Thangali, "The American Sign Language Lexicon Video Dataset," pp. 1–8, Jun. 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4563181

[60] C. Valli, Ed., *The Gallaudet Dictionary of American Sign Language*. Washington, DC: Gallaudet U. Press, 2006.

[61] "Zicheng Liu." [Online]. Available: http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/

[62] H. Nanda and K. Fujimura, "Visual tracking using depth data," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, june 2004, p. 37.

[63] PrimeSense, LTD, "OpenNI SDK | OpenNI," http://www.openni.org/openni-sdk/. [Online]. Available: http://www.openni.org/openni-sdk/

[64] ——, "NiTE 2.2.0.11 | OpenNI." [Online]. Available: http://www.openni.org/files/nite/

[65] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2821–2840, 2013.

[66] Microsoft.com, "Kinect for windows," 2015. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/default.aspx

[67] D. Bragg, K. Rector, and R. E. Ladner, "A user-powered american sign language dictionary," 2015.

[68] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos, "Dynamic affine-invariant shape-appearance handshape features and classification in sign language videos," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1627–1663, 2013.

[69] T. Pfister, J. Charles, and A. Zisserman, "Domain-adaptive discriminative one-shot learning of gestures," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 814–829.

[70] G. Pavlakos, S. Theodorakis, V. Pitsikalis, S. Katsamanis, and P. Maragos, "Kinect-based multimodal gesture recognition using a two-pass fusion scheme," in *Proc. Intl Conf. on Image Processing*, 2014.

[71] R. Elliott, H. Cooper, E.-J. Ong, J. Glauert, R. Bowden, and F. Lefebvre-Albaret, "Search-by-example in multilingual sign language databases," in *Proc. Sign Language Translation and Avatar Technologies Workshops*, 2011.

[72] Qt-project.org, "Qt project," 2015. [Online]. Available: http://qt-project.org/

[73] Opencv.org, "Opencv," 2015. [Online]. Available: http://opencv.org/

[74] Microsoft.com, "Developing with kinect," 2015. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/develop/

[75] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The american sign language lexicon video dataset," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on.* IEEE, 2008, pp. 1–8.

[76] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 12 1966.

[77] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[78] T. Darrell and A. Pentland, "Space-time gestures," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, Jun 1993, pp. 335–340.

[79] T. J. Darrell, I. A. Essa, and A. P. Pentland, "Task-specific gesture analysis in real-time using interpolated views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1236–1242, Dec 1996.

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

[81] C. Conly, Z. Zhang, and V. Athitsos, "An integrated rgb-d system for looking up the meaning of signs," in *Proceedings of the 8th ACM International Conference*

on PErvasive Technologies Related to Assistive Environments, ser. PETRA '15. New York, NY, USA: ACM, 2015, pp. 24:1–24:8.

[82] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[83] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[84] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer Berlin Heidelberg, 2012, pp. 5–13.

[85] A. D. Wilson and A. F. Bobick, "Learning visual behavior for gesture analysis," in *Computer Vision, 1995. Proceedings., International Symposium on*, Nov 1995, pp. 229–234.

[86] ——, "Hidden markov models." River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002, ch. Hidden Markov Models for Modeling and Recognizing Gesture Under Variation, pp. 123–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=505741.505748

[87] S. C. W. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 873–891, 2005. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2005.112

[88] ——, "Deciphering gestures with layered meanings and signer adaptation," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, May 2004, pp. 559–564.

[89] ——, "A new probabilistic model for recognizing signs with systematic modulations," in *Analysis and Modeling of Faces and Gestures, Third International Workshop, AMFG 2007, Rio de Janeiro, Brazil, October 20, 2007, Proceedings*, 2007, pp. 16–30.

[90] S. C. W. Ong, S. Ranganath, and Y. V. Venkatesh, "Understanding gestures with systematic variations in movement dynamics," *Pattern Recognition*, vol. 39, no. 9, pp. 1633–1648, 2006.

[91] ——, "Deciphering layered meaning in gestures," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 3, 2002, pp. 815–818 vol.3.

[92] M. Reyes, G. Domnguez, and S. Escalera, "Featureweighting in dynamic time-warping for gesture recognition in depth data," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 1182–1188.

[93] F. Yin, X. Chai, Y. Zhou, and X. Chen, "Weakly supervised metric learning towards signer adaptation for sign language recognition," in *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, 2015, pp. 35.1–35.12.

[94] A. Yao, L. V. Gool, and P. Kohli, "Gesture recognition portfolios for personalization," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1923–1930.

[95] M. Á. Bautista, A. Hernández-Vela, V. Ponce-López, X. Perez-Sala, X. Baró, O. Pujol, C. Angulo, and S. Escalera, "Probability-based dynamic time warping for gesture recognition on RGB-D data," in *Advances in Depth Image Analysis and Applications - International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers*, 2012, pp. 126–135.

[96] C. Conly, P. Doliotis, P. Jangyodsuk, R. Alonzo, and V. Athitsos, "Toward a 3d body part detection video dataset and hand tracking benchmark," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '13. New York, NY, USA: ACM, 2013, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/2504335.2504337

[97] "Dragon 13 NaturallySpeaking home." [Online]. Available: http://www.nuance.com/dragon/index.htm