Visual Recognition of American Sign Language Using Hidden Markov Models

by
Thad Eugene Starner

S.B., Computer Science
S.B., Brain and Cognitive Science
Massachusetts Institute of Technology, Cambridge MA
June 1991

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
at the
Massachusetts Institute of Technology
February 1995

Signature of Author ______________________________
Program in Media Arts and Sciences
20 January 1995

Certified by ______________________________
Alex Pentland
Head, Perceptual Computing Section
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ______________________________
Stephen A. Benton
Chairperson
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Visual Recognition of American Sign Language Using Hidden Markov Models

by
Thad Eugene Starner

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on January 20, 1995
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

Abstract

Using hidden Markov models (HMM's), an unobstrusive single view camera system is developed that can recognize hand gestures, namely, a subset of American Sign Language (ASL). Previous systems have concentrated on finger spelling or isolated word recognition, often using tethered electronic gloves for input. We achieve high recognition rates for full sentence ASL using only visual cues.

A forty word lexicon consisting of personal pronouns, verbs, nouns, and adjectives is used to create 494 randomly constructed five word sentences that are signed by the subject to the computer. The data is separated into a 395 sentence training set and an independent 99 sentence test set. While signing, the 2D position, orientation, and eccentricity of bounding ellipses of the hands are tracked in real time with the assistance of solidly colored gloves. Simultaneous recognition and segmentation of the resultant stream of feature vectors occurs five times faster than real time on an HP 735. With a strong grammar, the system achieves an accuracy of 97%; with no grammar, an accuracy of 91% is reached (95% correct).

Thesis Supervisor: Alex Pentland
Title: Head, Perceptual Computing Section, MIT Media Lab

This work was supported in part by the USAF Laboratory Graduate Fellowship program.

Visual Recognition of American Sign Language Using Hidden Markov Models

by
Thad Eugene Starner

The following people served as readers for this thesis:

Reader: ____________________________

Alex Pentland
Head, Perceptual Computing Section
Program in Media Arts and Sciences

Reader: ____________________________

Nathaniel I. Durlach
Senior Research Scientist
Electrical Engineering and Computer Science

Reader: ____________________________

Pattie Maes
Assistant Professor
Program in Media Arts and Sciences

I'd like to thank Alex Pentland, my advisor, for insisting over my objections that this project was possible within the bounds of a Master's thesis. His experienced insight into what is possible and worthwhile has guided me throughout my MIT career. I would also like to thank my readers, Nathaniel Durlach and Pattie Maes for coping with the changes over the past year.

Many thanks to the USAF Lab Graduate Fellowship program for funding my graduate studies.

Thanks also to John Makhoul, Rich Schwartz, Long Nguyen, Bruce Musicus, Paul Placeway, and numerous others in the Bolt, Barenek, and Newman Speech Group for giving me a good understanding of and intuition for modern HMM techniques. Along the same lines, thanks to Roz Picard and Alex Pentland for their lucid explanations of pattern recognition techniques which got me started in this field.

Thanks to Judy Bornstein for sharing her experience with ASL and for proofing this document.

Thanks to Mike P. Johnson, Ken Russell, and the IVE gang at the Media Lab for their contributions of code snippets, which made my work much easier. Thanks also to the Vision and Modeling Group, which has harbored me for many years as I've been "playing with the toys." It is my pleasure to work with such talented and exciting people.

A big thank-you to my parents, who have supported me without fail since I first discovered that I wanted to do research at MIT.

Last but not least, thanks to Tavenner Hall for keeping me sane through the last death throes of this thesis.

Contents

Chapter 1

Introduction

To date, computers have had very limited means of communicating with humans. Most common methods involve using a tethered device (keyboard, mouse, light pen, 3D tracker, etc.) that limit the user's freedom of motion. Furthermore, the expressiveness of these interactions has been very poor. With the advent of more powerful computers equipped with video, vision based interfaces are becoming more feasible. Enabling the computer to "see" its user allows for richer and more varied paradigms of man-machine interaction.

Recently, there has been a surge in interest in recognizing human hand gestures. While there are many interesting domains, one of the most structured sets of gestures are those belonging to sign language. In sign language, each gesture already has an assigned meaning (or meanings) and strong rules of context and grammar may be applied to make recognition tractable.

Most work on sign language recognition employs expensive wired "datagloves" that the user must wear [39]. In addition, these systems mostly concentrate on finger signing, where the user spells each word with hand signs corresponding to the letters in the alphabet [10]. However, most sign does not involve finger spelling but signs that represent whole words. This allows signed conversations to proceed at about the pace of spoken conversation.

In this paper an extensible system is described that uses a single color camera to track hands in real time and recognize sentences of American Sign Language (ASL) using hidden Markov models (HMM's). The hand tracking stage of the system does not attempt to produce a fine-grain description of hand shape; studies have shown that such detailed information may be unnecessary for humans to interpret sign language [28]. Instead, the tracking process produces only a coarse description of hand shape, orientation, and trajectory. The user is required to wear inexpensive colored gloves to facilitate the hand tracking frame rate and stability. This shape, orientation, and trajectory information is then input to an HMM for recognition of the signed words.

1.1 Applications

For many years the problem of continuous speech recognition has been a focus of research, with the goal of using speech as an interface. Similarly, if a full vocabulary recognition system for American Sign Language can be created, then ASL can be used in applications such as word processing, operating system control, etc. Perhaps the most promising application of an ASL recognition system is that of translation of ASL into written or spoken English. The translation problem involves more than recognizing signs, however; some level of grammar structure and meaning will have to be understood by the system to allow adequate translation.

Gestures are often made at points of stress in a conversation, when illustrating a motion, or when describing an object. In fact, research has been done on the language of these gestures [6]. By recognizing gestures made in conjunction with spoken language, a computer may be able to better understand the wishes of the user. If an ASL recognizer can be created, then similar technology may be applied to these conversational gestures as well.

Recently, the field of video annotation has gained popularity. Given a huge database of video footage, how are particular shots located? With a hand annotated database, a user can search the text hoping that the annotator attended what is desired. However, hand annotation is a time-consuming process. Instead, computer systems may be employed to annotate certain features of sequences. A human gesture recognition system adds another dimension to the types of features computers can automatically annotate.

1.2 Outline

Chapter 2 describes some attributes of American Sign Language and the scope of this thesis. Chapter 3 discusses previous work in related areas and develops the reasoning for choosing HMM's over other techniques. Details on the machine vision algorithms and the HMM training and recognition methods used are provided in Chapter 4. Chapter 5 describes the experiments performed and lists the results. An analysis of the results is provided in Chapter 6, and a summary and discussion of future work is included in Chapter 7.

Chapter 2

Problem Description

When focusing the techniques of machine vision on the human body, many diverse fields are addressed. Techniques from cognition, psychophysics, dynamics, photography, and athletics can all be applied to help constrain problem domains.

2.1 Analyzing Human Body Motion

Photography has been used to help understand human body motion for over a century [25]. More recently, computers have been added to perform more complex analysis. Athletic programs may use computer tracking systems and dynamics to help maximize the amount of effort their athletes can produce. Many of these systems use hand labeled data or wired sensor systems to produce the data. Fully reconstructing the motion of the human body requires a tremendous amount of data. Therefore, both natural constraints of the human body and simplifying assumptions are often used to curtail the amount of data needed for analysis.

Several systems that address whole body systems have been developed in the past. These include gait recognition and analysis systems [27, 14, 32, 42], ballet step recognition [5], body capture [1], real-time interface systems [38, 23], and numerous others. Greater accuracy and detail can be gained by focusing attention on the body part of interest. Recent experimentation with active "focus of attention" systems is attracting interest to this topic [9]. In the case of ASL, the hands and head are of the most interest.

2.2 American Sign Language

American Sign Language is the language of choice for most deaf people in the United States. It is part of the "deaf culture" and includes its own system of puns, inside jokes, etc. However, ASL is but one of the many sign languages of the world. A speaker of ASL would have trouble understanding the Sign Language of China much as an English speaker would Chinese. ASL also uses its own grammar instead of borrowing it from English. This grammar allows more flexibility of word placement and sometimes uses redundancy for emphasis. Another variant, English Sign Language has more in common with spoken English but is not as widespread in America. ASL consists of approximately 6000 gestures of common words with finger spelling used to communicate obscure words or proper nouns. Finger spelling uses one hand and 26 gestures to communicate the 26 letters of the alphabet; however, users prefer full word signs whenever possible since this allows sign language to approach or surpass the speed of conversational English.

Conversants in ASL will often describe a person, place, or thing and then point to a place in space to temporarily store that object for later reference [35]. For example, "the man with the green sweater," "the old grocery store," and "the red garage" might be signed and put into various positions in space. To then say "the man with the green sweater went to the old grocery store and the red garage" would involve pointing to the location of the man and then making the sign for "walk" to the position of the store and garage in turn. For the purposes of this thesis, this particular spatial aspect of ASL will be ignored.

ASL uses facial expressions to distinguish between statements, questions, and directives. The eyebrows are raised for a question, held normal for a statement, and furrowed for a directive. While there has been work in recognizing facial gestures [11], facial features will not be used to aid recognition in the task addressed.

Traditionally, there are three components of an ASL sign:

tabular (tab): The position of the hand at the beginning of the sign.

designator (dez): The hand shape of the hand at the beginning of the sign.

signation (sig): The action of the hand(s) in the dynamic phase of the sign.

In the Stokoe ASL dictionary [37], 12 tabulars, 19 designators, and 24 signations are distinguished. Even though both hands are used in ASL, only seven of the signations use two hands. Some signs depend on finger placement and movement to remove ambiguity, but many signs are distinct even when the fingers are ignored.

2.3 Goals

While the scope of this thesis is not to create a person independent, full lexicon system for recognizing ASL, a desired attribute of the system is extensibility towards this goal. Another goal is to allow the creation of a real-time system by guaranteeing each separate component (tracking, analysis, and recognition) runs in real-time. This demonstrates the possibility of a commercial product in the future, allows easier experimentation, and simplifies archiving of test data. "Continuous" sign language recognition of full sentences is desired to demonstrate the feasibility of recognizing complicated series of gestures. Of course, a low error rate is also a high priority.

Table 2.1: ASL Vocabulary Used

part of speech	*vocabulary*
pronoun	I you he we you(pl) they
verb	want like lose dontwant dontlike love pack hit loan
noun	box car book table paper pants bicycle bottle can wristwatch umbrella coat pencil shoes food magazine fish mouse pill bowl
adjective	red brown black gray yellow

In this recognition system, sentences of the form "personal pronoun, verb, noun, adjective, (the same) personal pronoun" are to be recognized. This sentence structure emphasizes the need for a distinct grammar for ASL recognition and allows a large variety of meaningful sentences to be randomly generated using words from each class. Table 2.3 shows the words chosen for each class. Six personal pronouns, nine verbs, twenty nouns, and five adjectives are included making the total lexicon number forty words. The words were chosen by paging through "A Basic Course in American Sign Language" by Humphries, Padden, and O'Rourke and selecting those which would provide coherent sentences when used to generate random sentences. At first a naive eye was used to avoid ambiguities in

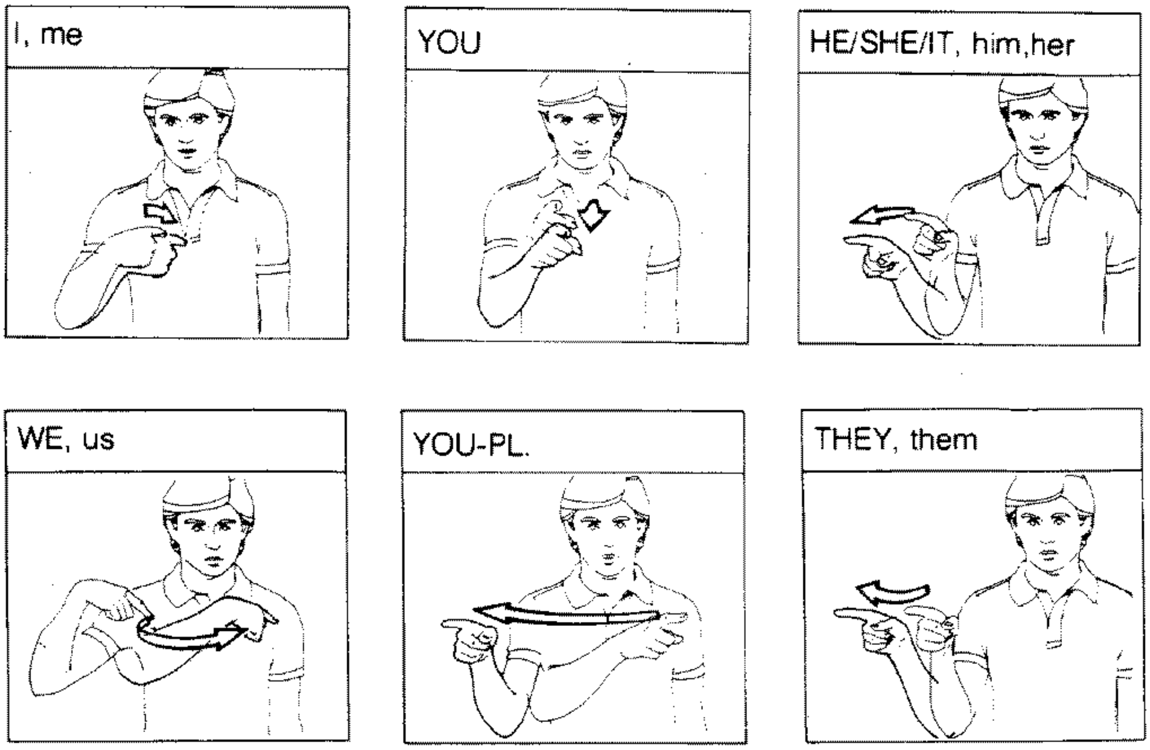

Figure 2-1: Pronouns.

the selected signs, but this was shortly subsumed by the coherency constraint. Figures 2-1 to 2-5 illustrate the signs selected (from [17]).

The process of creating a new recognition system often limits the amount of initial training that can be collected. False starts and complications sometimes require discarding of data, making the initial process extremely frustrating for the subject. Therefore, the author learned the necessary signs to provide the database.

Recognition will occur on sentences unseen by the training process. The task is to correctly recognize the words in the given sentence in order and without inserting any additional words. Error and accuracy will be measured as in the continuous speech recognition literature, incorporating substitution, insertion, and deletion errors.

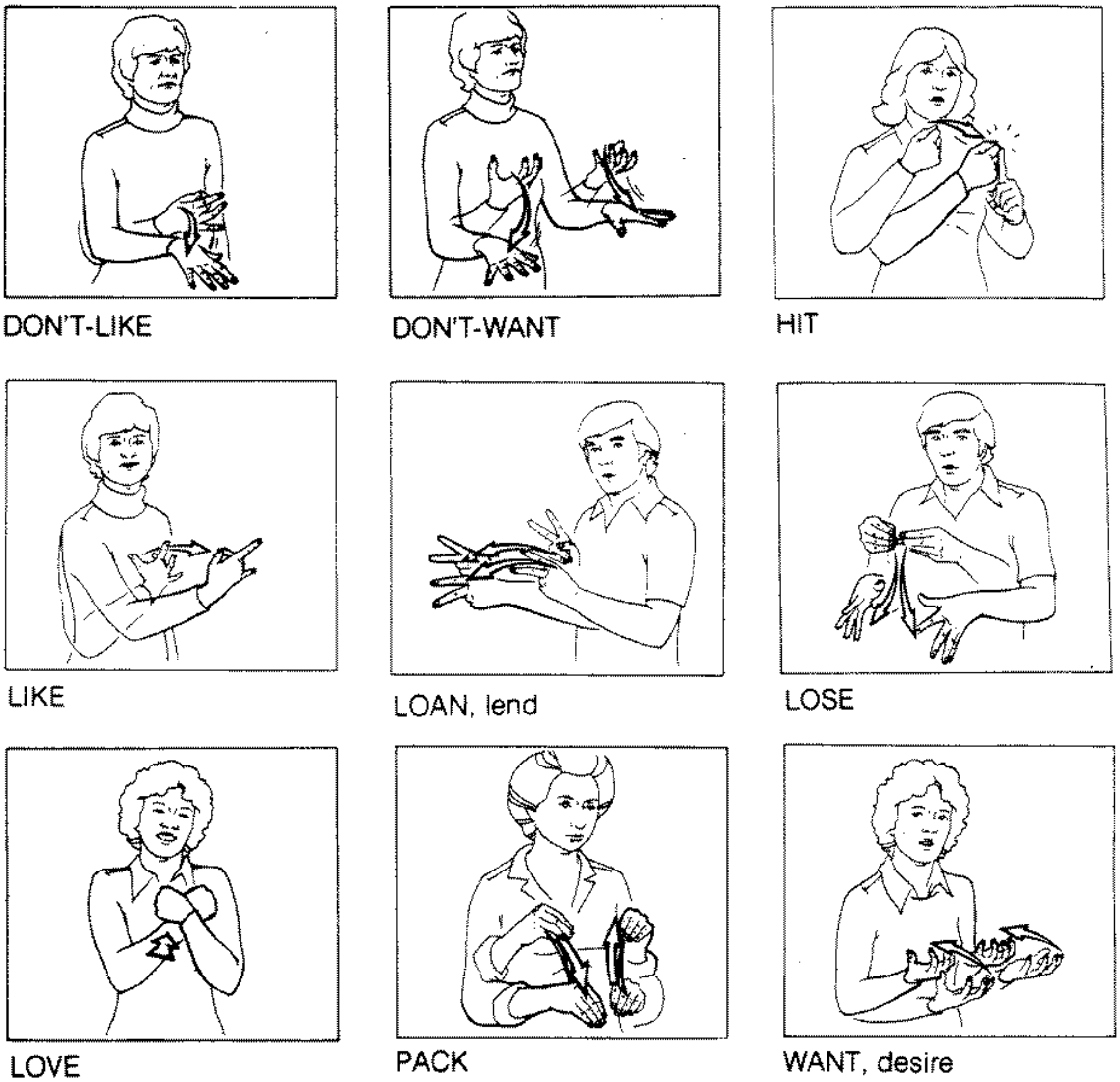

Figure 2-2: Verbs.

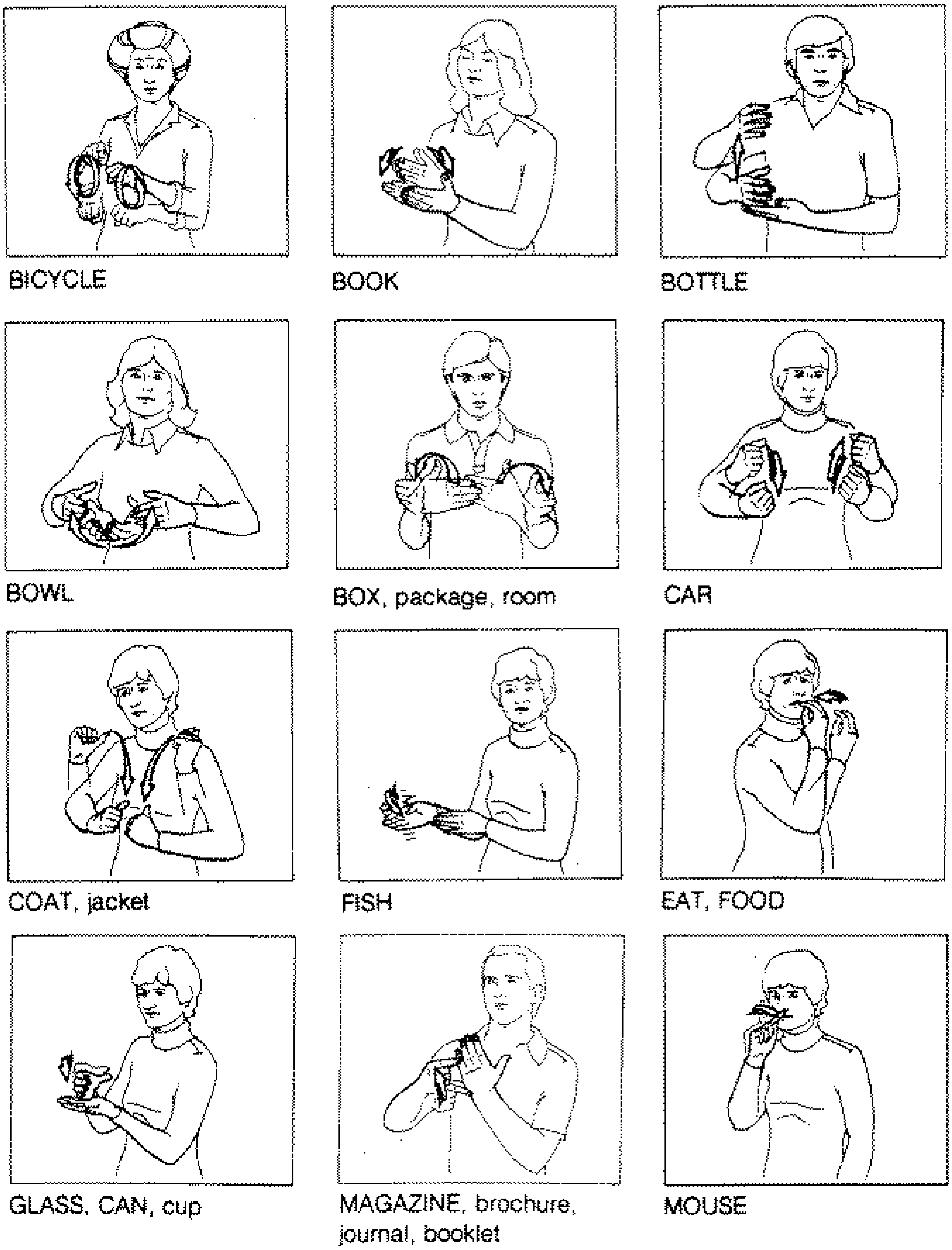

Figure 2-3: Nouns.

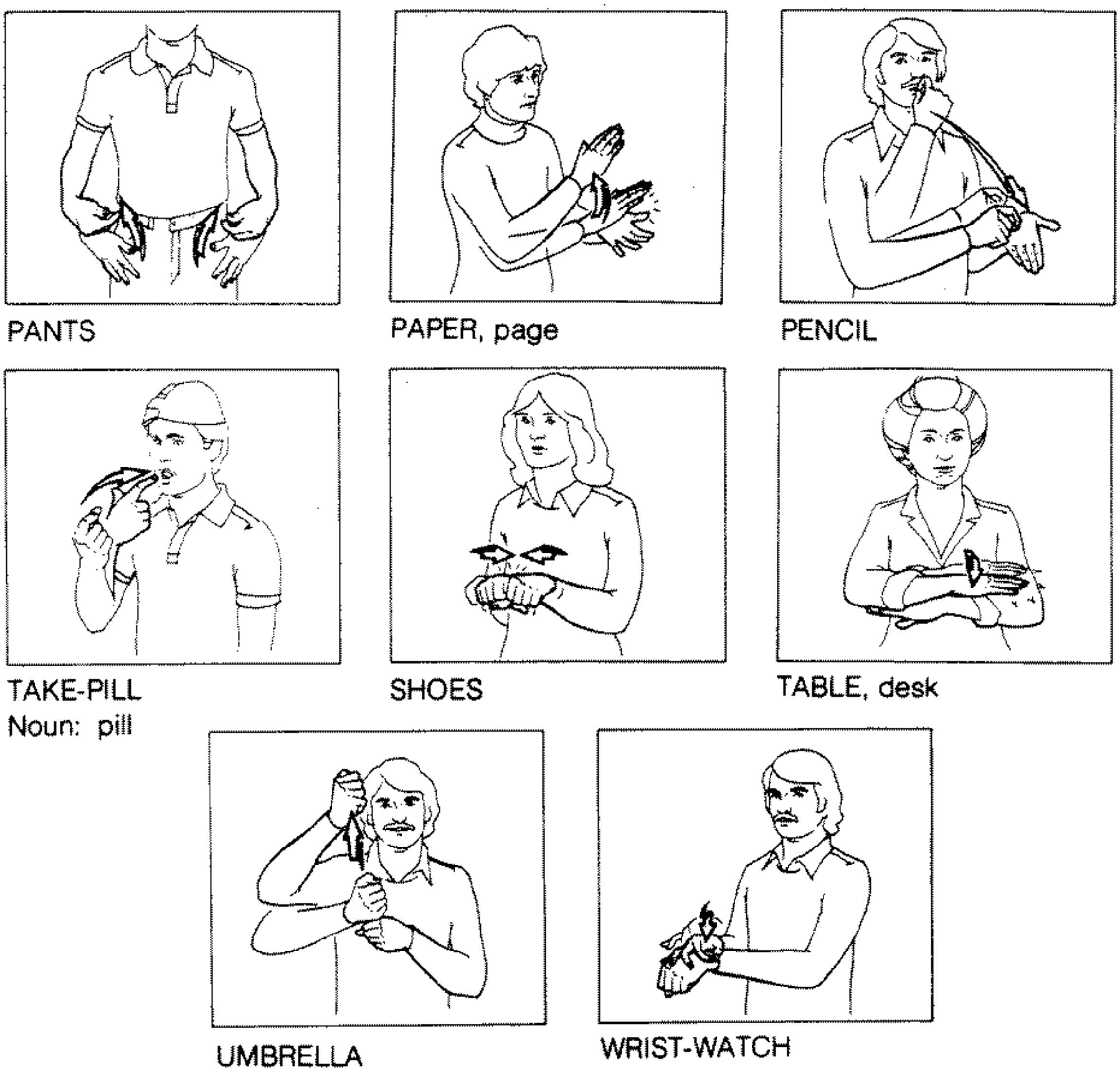

Figure 2-4: Nouns continued.

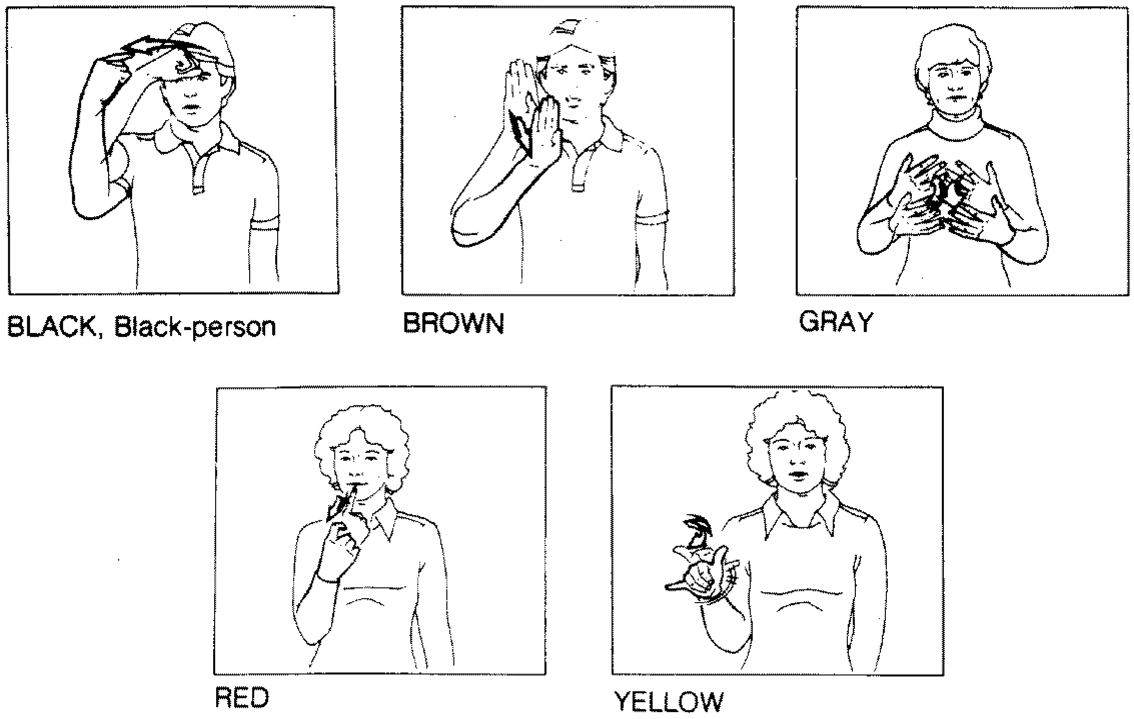

Figure 2-5: Adjectives.

Chapter 3

Background

Visual recognition of sign language requires two main components: hand tracking and pattern recognition. Machine vision and virtual environment research have provided several tools in addressing the former, and continuous speech recognition has provided an excellent development platform for the latter. Here, recent tracking systems are surveyed, previous sign language work is reviewed, and the benefits of applying HMM technology to machine recognition of ASL are discussed.

3.1 Hand Recovery

With "multimedia" computers being packaged with video cameras, interest in human gesture recognition has grown. A large variety of interfaces have been proposed, using video driven gestures for mouse control [12], full body interactions [19, 23, 5], expression tracking [11], conducting music [24], and electronic presentation [38].

Due to their expressiveness, the hands have been a point of focus for many gesture recognition systems. Tracking the natural hand in real time using camera imagery is difficult, but successful systems have been demonstrated in controlled settings. Freeman [12] has shown a hand tracker that can be used for navigating 3D worlds. A greyscale camera tracks the hand in a small area on a table and uses hand and finger position to control the direction of a virtual graphics camera.

Rehg and Kanade have shown a two camera system that can recover the full 27 degrees of freedom in a hand [31]. While successfully demonstrating tracking with limited motion, occlusion was not addressed. Furthermore, a simple background was required, which is often impossible when observing natural hand gestures.

Simpler, less constrained hand tracking systems have been created in a variety of environments. Krueger [19] has shown light table systems that tracked hands using single and multiple cameras aimed at the entire body. Maes *et al* [23] showed a similar system with one camera which allowed more arbitrary backgrounds to be used. Unfortunately, camera resolution often limits whole body systems to recovering just the position of the hands. When cameras are dedicated to the hands, more detail can be obtained. The "Hand Reader" by Suenaga *et al* [38] recovers 3D pointing information by dedicating two cameras, at close range, to the task. A limitation of such systems is that the working volume tends to be small. Also, the cameras for these systems tend to be obtrusive in that they are placed near the user. Longer focal length lenses may be used so that the cameras may be moved farther from the user, but then the space needed by such systems becomes prohibitive. Another solution is to monitor an entire room with a single fixed camera and use narrow field of view cameras mounted on servo platforms to direct attention to specific areas of interest [9]. This technique allows high detail and a wide range of motion, but suffers from the coupled motion problems of an active camera.

Tracking can often be simplified through using calibrated gloves or wired sensors. Dorner [10] uses a specially calibrated glove with different colors for each finger and the wrist and markers at each finger joint and tip. This, combined with Kalman filtering, simplifies occlusion problems and allows recovery of a detailed hand model through a wide range of motion. Datagloves by VPL are often used for sensing as well [24, 39, 4]. These systems offer precision, a relatively large range of motion (the sphere defined by the length of the tether), and very fast update rates at the expense of being wired to a sensing system.

3.2 Machine Sign Language Recognition

Excellent work has been done in support of machine sign language recognition. Sperling and Parish [35, 28] have done careful studies on the bandwidth necessary for a sign conversation using spatially and temporally subsampled images. Point light experiments (where "lights" are attached to significant locations on the body and just these points are used for recognition), have been carried out by Poizner *et al* [29] and Tartter and Knowlton [41]. Tartter's 27 light experiment (13 lights per hand plus one on the nose) showed that a brief conversation in ASL was possible using only these stimuli. Poizner's experiments used only 9 lights (on the head and each shoulder, elbow, wrist, and index finger) to convey limited information about ASL signs. These experiments suggest that ASL might be recognizable even in an impoverished environment.

Most of the above experiments studied ASL in context. However, most machine recognition systems to date have studied isolated and/or static gestures. In many cases these gestures are finger spelling signs, whereas everyday ASL uses word signs for speed.

Tamura and Kawasaki demonstrated an early image processing system which could recognize 20 Japanese signs based on matching cheremes [40]. A chereme is composed of the tab, dez, and sig as discussed earlier. Charayaphan and Marble [7] demonstrated a feature set that could distinguish between the 31 isolated ASL signs in their training set. Takahashi and Kishino in [39] discuss a Dataglove-based system that could recognize 34 of the 46 Japanese kana alphabet gestures (user dependent) using a joint angle and hand orientation coding technique. From their paper, it seems the test user made each of the 46 gestures 10 times to provide data for principle component and cluster analysis. A separate test set was created from five iterations of the alphabet by the user, with each gesture well separated in time. While these systems are technically interesting, they suffer from a lack of training and have limited expandability beyond their sample domains.

3.3 Previous Use of Hidden Markov Models in Gesture Recognition

While the continuous speech recognition community adopted HMM's many years ago, these techniques are just now entering the vision community. Most early work was limited to handwriting recognition [21, 26]. More recently, He and Kundu [13] report using continuous density HMM's to classify planar shapes. Their method segmented closed shapes and exploited characteristic relations between consecutive segments for classification. The algorithm was reported to tolerate shape contour perturbation and some occlusion.

Another early effort by Yamato *et al* uses discrete HMM's to successfully recognize image sequences of six different tennis strokes among three subjects. This experiment is significant because it used a 25x25 pixel quantized subsampled camera image as a feature vector. Even with such low-level information, the model could learn the set of motions to perform respectable recognition rates.

Schlenzig *et al* [33] also use hidden Markov models for visual gesture recognition. The gestures are limited to "hello," "good-bye," and "rotate". The authors report "intuitively" defining the HMM associated with each gesture and imply that the normal Baum-Welch re-estimation method was not implemented. However, this study shows the continuous gesture recognition capabilities of HMM's by recognizing gesture sequences.

Several vision systems have been developed with technology closely related to HMM methodology. Darrell [8] uses the dynamic time warping method to recognize gestures ("hello" and "good-bye") through time. Siskind and Morris [34] argue that event perception requires less information and may be an easier problem than object recognition. To this end they have constructed a maximum likelihood framework using methods similar to those used in hidden Markov model training to recognize the events "pick up," "put down," "drop," and "fall" from edge-detected movies of these actions. While these projects do not use HMM's *per se*, they set the stage for the idea that visual information can be modeled and used for recognition through time, much like speech recognition. In fact, Siskind and Morris make similar comparisons as those below between continuous speech recognition and vision.

3.4 Use of HMM's for Recognizing Sign Language

While the above studies show some promise for using HMM's in vision, what evidence is there that HMM's can be eventually used to address the full ASL recognition problem? The answer to that question lies in the comparison of the ASL recognition domain to the continuous speech recognition domain where HMM's have become the technology of choice.

Sign language and continuous speech share many common characteristics. Sign language can be viewed as a signal (position, shape, orientation of the hands, etc.) over time, just like speech. Silences in both speech and ASL are relatively easy to detect (hesitating between signs will be considered equivalent to stuttering or "ums" in speech), and all of the information needed to specify a fundamental unit in both domains is given contiguously in a finite time period. The onset and offset paths of a sign depend on the temporarily neighboring signs. Correspondingly, spoken phonemes change due to coarticulation in speech. In both domains, the basic units combine to form more complex wholes (words to phrases in sign and phonemes to words in speech). Thus, language modeling can be applied to improve recognition performance for both problems.

In spite of the above similarities, sign language recognition has some basic differences from speech recognition. Unlike speech where phonemes combine to create words, the fundamental unit in much of ASL is the word itself. Thus, there is not as much support for individual word recognition in sign as there is in speech. Also, the fundamental unit in sign can switch suddenly (for example, changing into finger spelling for proper nouns). Furthermore, the grammar in ASL is significantly different than that of English speech (ASL is strongly influenced by French). However, even given these differences, there seems a strong likelihood that HMM's should also apply to sign language recognition.

Hidden Markov models have intrinsic properties which make them very attractive for ASL recognition. All that is necessary for training, except when using an optional bootstrapping process, is a data stream and its transcription (the text matching the signs). The training process can automatically align the components of the transcription to the data. Thus, no special effort is needed to label training data. The segmentation problem, as often seen in handwriting research, can be avoided altogether [36].

Recognition is also performed on a continuous data stream. Again, no explicit segmentation is necessary. The segmentation of sentences into words occurs naturally by incorporating the use of a lexicon and a language model into the recognition process. The result is a text stream that can be compared to a reference text for error calculation. Consequently sign language recognition seems an ideal machine vision application of HMM technology. The problem domain offers the benefits of scalability, well defined meanings, a pre-determined language model, a large base of users, and immediate applications for a recognizer.

Finally, HMM methodology is related to techniques that have been used previously in vision with success. Dynamic time warping, expectation maximization, Q-learning, neural nets, and several other traditional pattern recognition techniques resemble portions of the modeling and recognition processes. The major advantage HMM's have over these other techniques is the ability to selectively, knowledgeably, and scalably tailor the model to the task at hand.

Chapter 4

Tracking and Modeling Gestures Using Hidden Markov Models

As shown in the last section, using computer vision to observe human motion is becoming a rich and diverse field. Traditionally, HMM's have been the domain of speech recognition, where a very rich system of modeling has evolved. In the following sections the vision methods used to track the hands are described, and the basics of HMM's are applied to ASL recognition.

4.1 Hidden Markov Modeling

While a substantial body of literature exists on HMM technology [43, 16, 30, 2, 20], this section modifies a traditional discussion of the algorithms so as to provide the perspective used for recognizing sign language. A simplistic example develops the fundamental theory in training and testing of a discrete HMM which is then generalized to the continuous density case used in the experiments. For broader discussion of the topic, [16] is recommended.

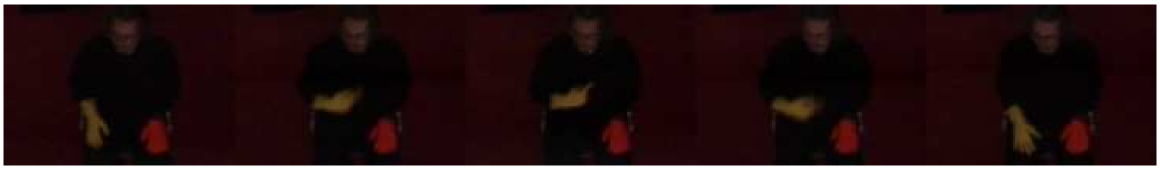

Figure 4-1: The ASL sign for "I."

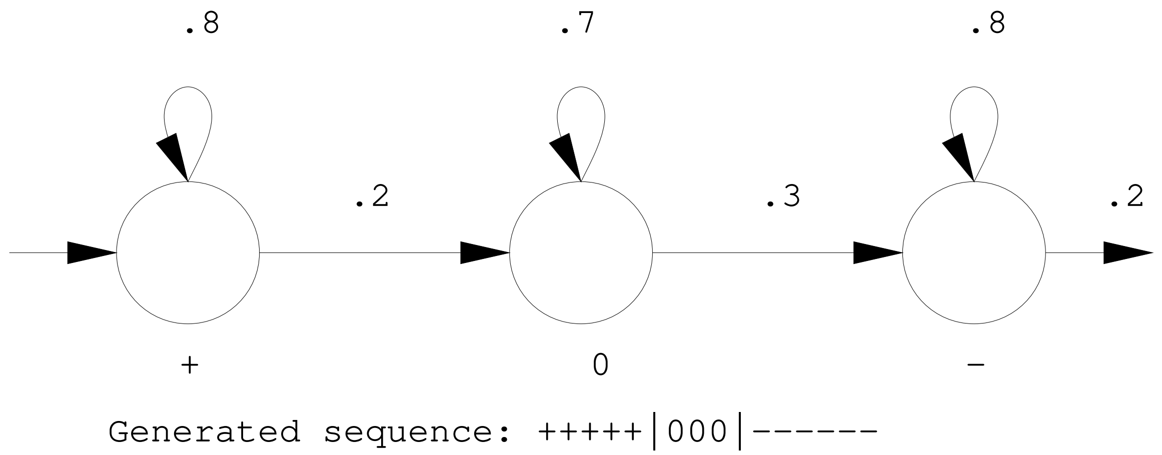

Figure 4-2: A sample HMM topology with transition probabilities. Note that generated sequence can be divided into the states which produced each section.

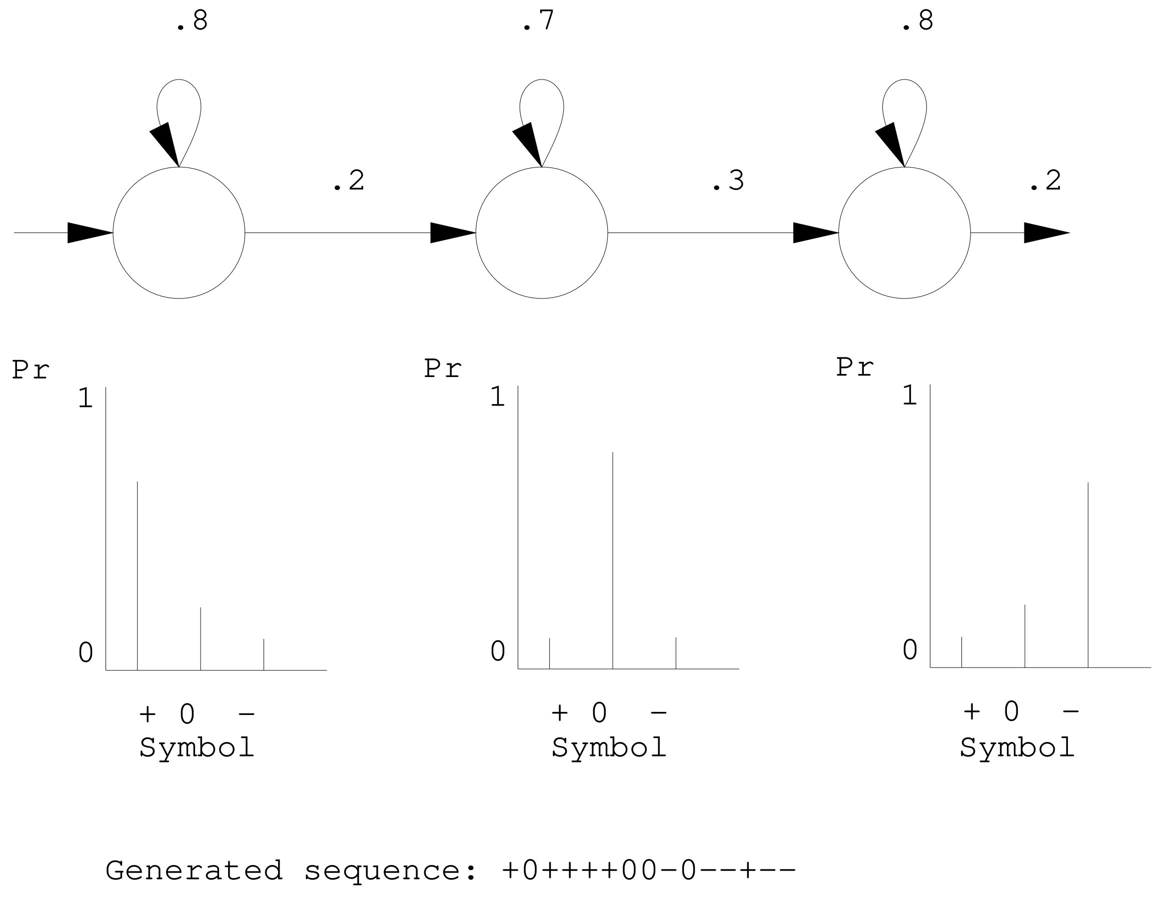

Figure 4-3: The same 3 state topology with output probabilities added. Now the state sequence can no longer be precisely recovered.

A time domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the jth most recent events. If the current event depends solely on the most recent past event, then the process is a first order Markov process. While the order of words in American Sign Language is not truly a first order Markov process, it is a useful assumption when considering the positions and orientations of the hands of the signer through time. Consider the hand actions for the isolated gesture "I" (Figure 4-1). These actions might be separated into the onset movement of the hand to the chest, pointing at the chest, and the offset movement of the hand back to a rest position. If the y movement of the right hand is used as a feature in recognizing these actions (say, at 5 frames/sec), the onset, pointing, and offset actions would correspond to a predominantly positive motion from the floor (+), a relatively stable period (0), and a predominantly negative motion respectively (-). Consider the onset, pointing, and offset actions to be "states" in a Markov model of the motions involved in the sign "I" (see Figure 4-2). Note that each state might transition back to itself. This corresponds to the inherent time aspect of the separate actions of the gesture. For example, the onset action may take a second, corresponding to four transitions (5 frames/sec) of the onset state back to itself. Similarly, the pointing and offset actions may take varying amounts of time to complete. This is reflected in the transition probabilities shown in Figure 4-2. In this simple example, associated with each state are the data of positive, negative, or no y movement. Thus, the model can be used to generate appropriate y movement for the gesture "I". However, in real life the motions for the gesture "I" are not deterministic. Due to possible indecision and wavering of the hand the actual y motion of the right hand might consist of varying proportions of positive, negative, and no motion during any given action. To reflect this, the model is changed to that in Figure 4-3 where the output of each state is a discrete probability distribution (output probability) of the three possible y classes. Now, if the model is used to generate appropriate y motion it is not obvious from which state each y movement is generated. This attribute is the reason for the word "hidden" in hidden Markov models; the state sequence is not obvious from the generated set of motions. This example demonstrates one other assumption for first order HMM's, that the output probability depends only on the current state and not on how or when the state was entered.

The previous paragraph described how to use HMM's to generate a sequence of symbols that statistically resembles the data that the modeled process might produce. For the rest of this section, this idea is turned around to show how HMM's can be used to recognize a similar string of symbols (which will now be called observations). In addition, the algorithms needed to train a set of HMM's are described.

In order to proceed, a clear standard for notation is needed. Below is a list of symbols that will be used in this discussion. The meaning for some of these variables will become clearer in context, but the reader is urged to gain some familiarity with them before continuing.

T: the number of observations.

N: number of states in the HMM.

L: distinct number of possible observations (in this example three: +, 0, -).

s: a state. For convenience (and with regard to convention in the HMM literature), state i at time t will be denoted as $s_t = i$.

S; the set of states. S_I and S_F will be used to denote the set of initial and final states respectively.

O_t: an observation at time t.

$\mathbf{O}$: an observation sequence $O_1, O_2, ...O_T$.

v: a particular type of observation. For example, v_1, v_2, and v_3 might represent +, 0, and - y motion respectively.

a: state transition probability. a_{ij} represents the transition probability from state i to state j.

A: the set of state transition probabilities.

b: state output probability. $b_j(k)$ represents the probability of generating some discrete symbol v_k in state j.

B: the set of state output probabilities.

π: initial state distribution.

λ: a convenience variable representing a particular hidden Markov model. λ consists of A, B, and π.

α: the "forward variable," a convenience variable. $\alpha_t(i)$ is the probability of the partial observation sequence to time t and state i, which is reached at time t, given the model λ. In notation, $\alpha_t(i) = Pr(O_1, O_2, ..., O_t, s_t = i|\lambda)$.

β: the "backward variable," a convenience variable. Similar to the forward variable, $\beta_t(i) = Pr(O_{t+1}, O_{t+2}, ..., O_T|s_t = i, \lambda)$, or the probability of the partial observation sequence from $t+1$ to the final observation T, given state i at time t and the model λ.

γ: generally used for *a posterior* probabilities. $\gamma_t(i,j)$ will be defined as the probability of a path being in state i at time t and making a transition to state j at time $t+1$, given the observation sequence and the particular model. In other words, $\gamma_t(i,j) = Pr(s_t = i, s_{t+1} = j|\mathbf{O}, \lambda)$. $\gamma_t(i)$ will be defined as the posterior probability of being in state i at time t given the observation sequence and the model, or $\gamma_t(i) = Pr(s_t = i|\mathbf{O}, \lambda)$.

While a few other variables will be introduced in the description of the following algorithms, the above variables are typically found in descriptions of work in the field.

There are three key problems in HMM use. These are the evaluation problem, the estimation problem, and the decoding problem. The evaluation problem is that given an observation sequence and a model, what is the probability that the observed sequence was generated by the model ($Pr(\mathbf{O}|\lambda)$)? If this can be evaluated for all competing models for an observation sequence, then the model with the highest probability can be chosen for recognition.

$Pr(\mathbf{O}|\lambda)$ can be calculated several ways. The naive way is to sum the probability over all the possible state sequences in a model for the observation sequence:

$$Pr(\mathbf{O}|\lambda) = \sum_{allS} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t)$$

The initial distribution π_{s_1} is absorbed into the notation for $a_{s_0}s_1$ for simplicity in this discussion. The above equation can be better understood by ignoring the outside sum and product and setting $t = 1$. Assuming a particular state sequence through the model and the observation sequence, the inner product is the probability of transitioning to the state at time 1 (in this case, from the initial state) times the probability of observation 1 being output from this state. By multiplying over all times 1 through T, the probability that the state sequence S and the observation sequence $\mathbf{O}$ occur together is obtained. Summing this probability for all possible state sequences S produces $Pr(\mathbf{O}|\lambda)$. However, this method is exponential in time, so the more efficient forward-backward algorithm is used in practice.

The forward variable has already been defined above. Here its inductive calculation, called the forward algorithm, is shown (from [16]).

- $\alpha_1(i) = \pi_i b_i(O_1)$, for all states i (if $i \epsilon S_I, \pi_i = \frac{1}{n_I}$;otherwise $\pi_i = 0$)

- Calculating $\alpha()$ along the time axis, for $t = 2, ..., T$, and all states j, compute

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_{ij}] b_j(O_t)$$

- Final probability is given by

$$Pr(\mathbf{O}|\lambda) = \sum_{i \epsilon S_F} \alpha_T(i)$$

The first step initializes the forward variable with the initial probability for all states, while the second step inductively steps the forward variable through time. The final step gives the desired result $Pr(\mathbf{O}|\lambda)$, and it can be shown by constructing a lattice of states and transitions through time that the computation is only order $O(N^2T)$.

Another way of computing $Pr(\mathbf{O}|\lambda)$ is through use of the backward variable β, as already defined above, in a similar manner.

- $\beta_T(i) = \frac{1}{N_F}$, for all states $i \epsilon S_F$; otherwise $\beta_T(i) = 0$

- Calculating $\beta()$ along the time axis, for $t = T-1, T-2, ..., 1$ and all states j, compute:

$$\beta_t(j) = \sum_i a_{ji} b_i(O_{t+1}) \beta_{t+1}(i)$$

- Final probability is given by:

$$Pr(\mathbf{O}|\lambda) = \sum_{i \epsilon S_I} \pi_i b_i(O_1)\beta_1(i)$$

The estimation problem concerns how to adjust λ to maximize $Pr(\mathbf{O}|\lambda)$ given an observation sequence $\mathbf{O}$. Given an initial model, which can have flat probabilities, the forward-backward algorithm allows us to evaluate this probability. All that remains is to find a method to improve the initial model. Unfortunately, an analytical solution is not known, but an iterative technique can be employed.

Referring back to the definitions and the gesture "I" example earlier, it can be seen that

$$\sum_{t \epsilon O_t = +} \gamma_t(1)$$

is the expected number of times "+" should occur (given the observation and model) during a typical onset action of the gesture "I." The total expected motions (+, 0, -) given the model and observation during the onset action is

$$\sum_{t=1}^{T} \gamma_t(1)$$

Note that now, from the actual evidence and these two calculations, a new estimate for the respective output probability (for up in the onset gesture) can be assigned. Generalizing for a new estimate of a given output probability,

$$\bar{b}_j(k) = \frac{\sum_{t \epsilon O_t = v_k} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Similarly, the evidence can be used to develop a new estimate of the probability of a state transition. Thus,

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

Initial state probabilities can also be re-estimated through the formula

$$\bar{\pi}_i = \gamma_1(i)$$

Thus all the components of λ, namely A, B, and π can be re-estimated. Since either the forward or backward algorithm can be used to evaluate $Pr(\mathbf{O}|\bar{\lambda})$ versus the previous estimation, the above technique can be used iteratively to converge the model to some limit. While the technique described only handles a single observation sequence, it is easy to extend to a set of observation sequences. A more formal discussion can be found in [16, 2, 43].

While the estimation and evaluation processes described above are sufficient for the development of an HMM system, the Viterbi algorithm provides a quick means of evaluating a set of HMM's in practice as well as providing a solution for the decoding problem. In decoding, the goal is to recover the state sequence given an observation sequence. The Viterbi algorithm can be viewed as a special form of the forward-backward algorithm where only the maximum path at each time step is taken instead of all paths. This optimization reduces computational load and additionally allows the recovery of the most likely state sequence. The steps to the Viterbi are

- Initialization. For all states i, $\delta_1(i) = \pi_i b_i(O_1)$; $\psi_i(i) = 0$
- Recursion. From $t = 2$ to T and for all states j, $\delta_t(j) = Max_i[\delta_{t-1}(i)a_{ij}]b_j(O_t)$; $\psi_t(j) = argmax_i[\delta_{t-1}(i)a_{ij}]$
- Termination. $P = Max_{s\epsilon S_F}[\delta_T(s)]$; $s_T = argmax_{s\epsilon S_F}[\delta_T(s)]$
- Recovering the state sequence. From $t = T-1$ to 1, $s_t = \psi_{t+1}(s_{t+1})$

In many HMM system implementations, the Viterbi algorithm is used for evaluation at recognition time. Note that since Viterbi only guarantees the maximum of $Pr(\mathbf{O}, S|\lambda)$ over all S (as a result of the first order Markov assumption) instead of the *sum* over all possible state sequences, the resultant scores are only an approximation. For example, if there are two mostly disjoint state sequences through one model with medium probability and one state sequence through a second model with high probability, the Viterbi algorithm would favor the second HMM over the first. However, [30] shows that the probabilities obtained from both methods may be typically very close.

In practice, the Viterbi algorithm may be modified with a limit on the lowest numerical value of the probability of the state sequence, which in effect causes a beam search of the space. While this modification no longer guarantees optimality, a considerable speed increase may be obtained. Furthermore, to aid in estimation, the Baum-Welch algorithm may be manipulated so that parts of the model are held constant while other parts are trained.

To date, the example using the y motion in the gesture "I" assumed quantization of the motion into three classes +,0, and -. It is easy to see how, instead of quantizing, the actual probability density for y motion might be used. However, the above algorithms must be modified to accept continuous densities. The efforts of Baum, Petrie, Liporace, and Juang [3, 2, 22, 18] showed how to generalize the Baum-Welch, Viterbi, and forward-backward algorithms to handle a variety of characteristic densities. In this context, however, the densities will be assumed to be Gaussian. Specifically,

$$b_j(O_t) = \frac{1}{\sqrt{(2\pi)^n|\sigma_j|}} e^{\frac{1}{2}(O_t-\mu_j)'\sigma_j^{-1}(O_t-\mu_j)}$$

Initial estimations of μ and σ may be gotten by dividing the evidence evenly among the states of the model and calculating the mean and variance in the normal way.

$$\mu_j = \frac{1}{T}\sum_{t=1}^{T} O_t$$

$$\sigma_j = \frac{1}{T}\sum_{t=1}^{T}(O_t - \mu_j)(O_t - \mu_j)'$$

Whereas flat densities were used for the initialization step before, here the evidence is used. Now all that is needed is a way to provide new estimates for the output probability. We wish to weight the influence of a particular observation for each state based on the likelihood of that observation occurring in that state. Adapting the solution from the discrete case yields

$$\bar{\mu_j} = \frac{\sum_{t=1}^{T} \gamma_t(j) O_t}{\sum_{t=1}^{T} \gamma_t(j)}$$

and

$$\bar{\sigma_j} = \frac{\sum_{t=1}^{T} \gamma_t(j)(O_t - \bar{\mu_j})(O_t - \bar{\mu_j})^t}{\sum_{t=1}^{T} \gamma_t(j)}$$

In practice, μ_j is used to calculate $\bar{\sigma_j}$ instead of the re-estimated $\bar{\mu_j}$ for convenience. While this is not strictly proper, the values are approximately equal in contiguous iterations [16] and seem not to make an empirical difference [43]. Since only one stream of data is being used and only one mixture (Gaussian density) is being assumed, the algorithms above can proceed normally incorporating these changes for the continuous density case.

4.2 Feature Extraction Given Binarized Images of the Hands

In the previous discussion, the y motion of the right hand, a scalar quantity, was used to demonstrate the mathematics behind continuous density HMM's. However, there is no factor excluding the use of vectors (in fact, the equations are written for vector form). Feature vectors will simply result in multi-dimensional Gaussian densities. Given this freedom, what features should be used to recognize sign language?

Previous experience has shown that starting simple and evolving the feature set is often best [36]. Since finger spelling is not being allowed and there are few ambiguities in the test vocabulary based on individual finger motion, a relatively coarse tracking system may be used. Based on previous work [23], it was assumed that a system could be designed to separate the hands from the rest of the scene (explained in the next section). Traditional vision algorithms could then be applied to the binarized result. Besides the position of the hands, some concept of the shape of the hand and the angle of the hand relative to horizontal seemed necessary. Thus, an eight element feature vector consisting of each hand's x and y position, angle of axis of least inertia, and eccentricity of bounding ellipse was decided upon. The eccentricity of the bounded ellipse was found by determining the ratio of the square roots of the eigenvalues that correspond to the matrix

$$\begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}$$

where a, b, and c are defined as

$$a = \int\int_{I'} (x')^2 dx' dy'$$

$$b = \int\int_{I'} x' y' dx' dy'$$

$$c = \int\int_{I'} (y')^2 dx' dy'$$

(x' and y' are the x and y coordinates normalized to the centroid)

The axis of least inertia is then determined by the major axis of the bounding ellipse, which corresponds to the primary eigenvector of the matrix [15]. Figure 4-4 demonstrates bounding ellipses fitted to the images of the hands. Note the 180 degree ambiguity in the angle of the ellipses. To address this problem, the angles were only allowed to range from -90 to +90 degrees.

4.3 Recovering the Hands from Video

Since real-time recognition is a goal, several compromises were made. The subject wears distinctly colored gloves on each hand (a yellow glove for the right hand and an orange glove for the left) and sits in a chair before the camera (see Figure 4-5). Figure 4-6 shows the view from the camera's perspective and gives an impression of the quality of video that is used. Color NTSC composite video is captured and analyzed at a constant 5 frames per second at 320 by 243 pixel resolution on a Silicon Graphics Indigo 2 with Galileo video board. To initially find each hand, the algorithm scans the image until it finds a pixel of the appropriate color. Given this pixel as a seed, the region is grown by checking the eight nearest neighbors for the appropriate color. Each pixel checked is considered to be part of the hand. This, in effect, performs a simple morphological dilation upon the resultant image that helps to prevent edge and lighting aberrations. The centroid is calculated as

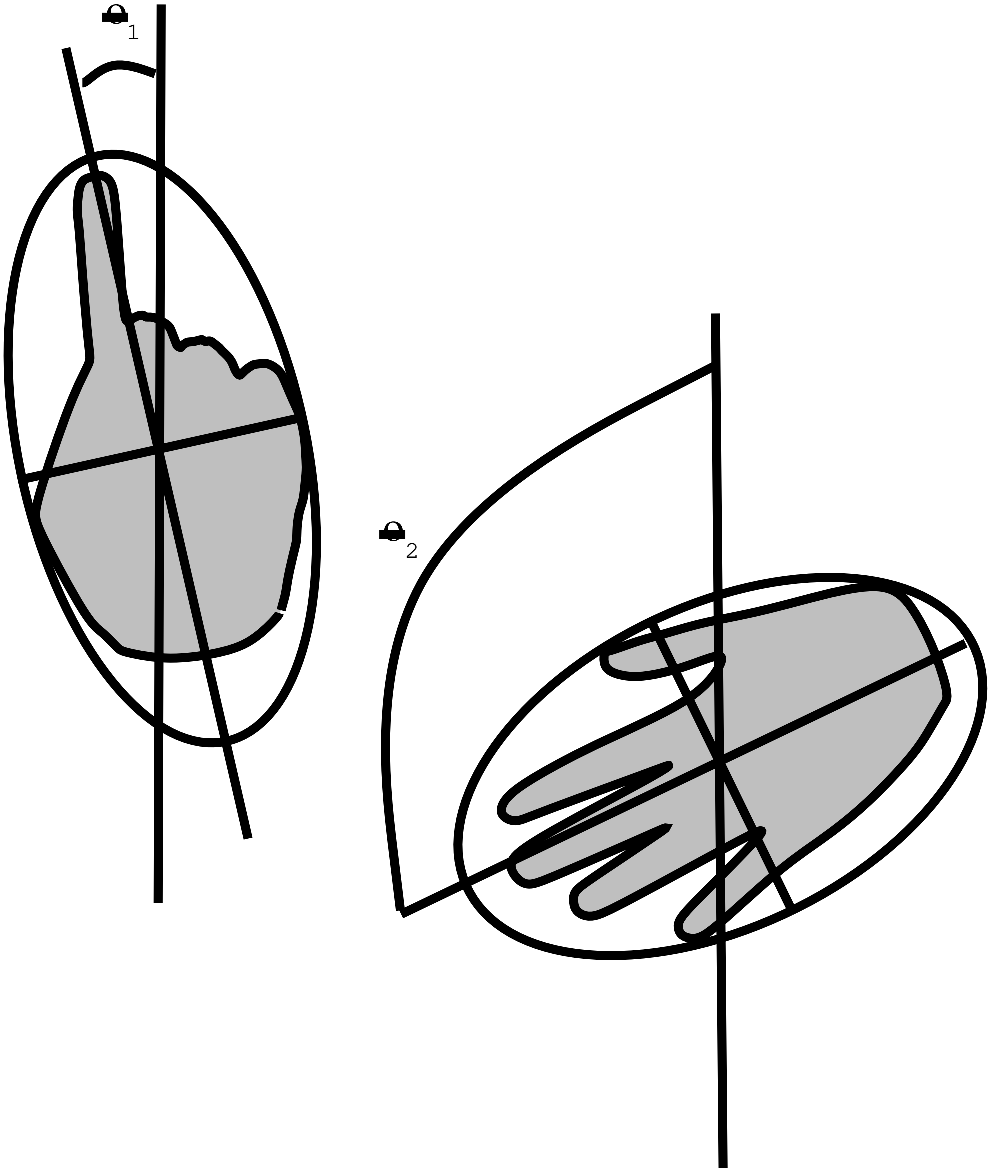

Figure 4-4: Bounding ellipses generated by tracking code.

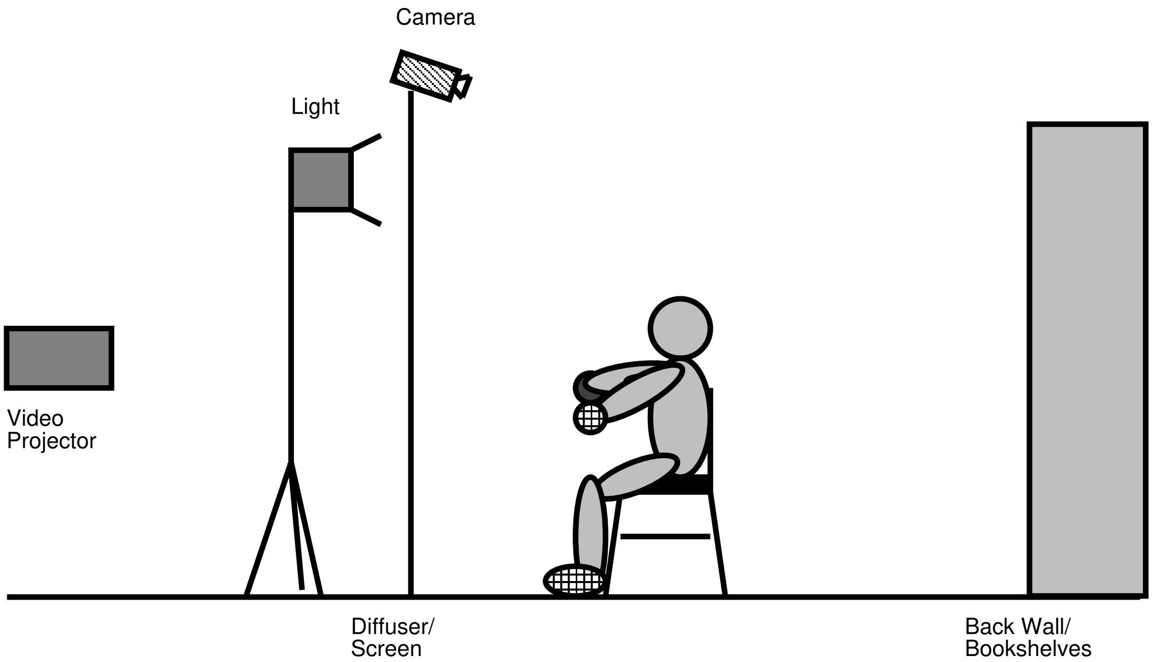

Figure 4-5: Tracking environment.

a by-product of the growing step and is stored as the seed for the next frame. Given the resultant bitmap and centroid, second moment analysis is performed as described earlier.

4.4 Selecting an HMM Topology

Previously, a 3 state model was used to represent the gesture "I." While this seems sufficient to model a simple gesture, such a model will not do as well in general. For more complex signs such as "table" which involves repetitive motion (onset; up, down, up motion of the right hand patting the left forearm; and offset), more states are necessary. In fact, a particular sign requires a different number of states if the gesturer is allowed to abbreviate the sign. However, through use of skip transitions, where the model has a certain probability of skipping part of the modeled sign, abbreviated signs can be accommodated (see Figure 4-7).

Skip transitions can also be used to avoid the task of determining the topology of each sign. For example, the model in Figure 4-7 can be trained to accommodate a three state sign like "I" through use of the skip states as well as a five state sign like "table," where the skip transitions would have low weights. Thus, once the minimum and maximum number

Figure 4-6: View from the tracking camera.

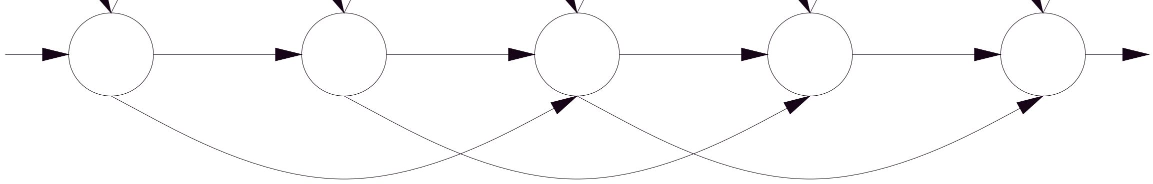

Figure 4-7: A more generalized topology

of states required is determined, a common topology may be used for all signs without too much loss of power (an added benefit is ease of coding).

4.5 Training an HMM network

When using HMM's to recognize strings of data, such as continuous speech, cursive handwriting, or ASL sentences, several methods can be brought to bear for training and recognition. Generally, individual models are concatenated together to model the larger language structures. Continuous speech recognition efforts have particularly advanced this field.

In speech, the fundamental unit (at least for these purposes) is the phoneme. Models for the individual phonemes can be trained separately, but this training is of limited utility if the goal is to recognize continuous sentences. In particular, the "co-articulation" effects of several phonemes spoken together may cause phonemes taken from continuous speech to differ significantly from phonemes spoken in isolation. An initial solution to this problem is to train on phonemes that have been manually segmented from continuous speech. When first addressing a task, manual segmentation is often worthwhile, especially when there is a small amount of training data and statistics on the field are not generally available. However, manual segmentation often has error and ambiguity. Furthermore, since the goal is recognition and not segmentation, it does not matter if the final system uses co-articulation to help recognize the phonemes. In fact, co-articulation is beneficial in that it provides context for recognizing the words of the sentence. To take advantage of this, two forms of context dependent training are used, embedded and context training.

Embedded training addresses the issue of segmentation. While initial training of the models might rely on manual segmentation or dividing the evidence evenly among the models, embedded training trains the models *in situ* and allows these boundaries to shift through a probabilistic entry into the initial states of each model [43].

Context training uses the co-occurrence of two or more fundamental units to allow recognition of blocks of units, which have more evidence than single units alone. In speech recognition, two and three phoneme blocks (biphones and triphones) are generally used. Note that recognition of these blocks might violate the first-order Markov assumption that was made earlier. However, by unrolling these blocks in the Viterbi process, the first-order assumption can be preserved.

A final use of context in speech recognition is on the word level. Statistical grammars relating the probability of the co-occurrence of two or more words can be used to weight the recognition process. Grammars that associate two words are called bigrams, whereas grammars that associate three words are called trigrams.

In ASL, the fundamental unit is the sign. Since most signs represent whole words, context training occurs at the sentence level. In fact, three sign contexts (trisines) might represent most of a sentence. With finger spelling however, a more direct relation of sign to phoneme, word to word, and sentence to sentence can be established. This may present a challenge in higher level language modeling in the future, but a simple solution is simply to treat finger spelling as if it was on a word level. Statistical grammars may have a place in helping to merge these two levels of signing. With exclusive word signing, grammars are still useful. Generally, context training occurs at the model level using the training data provided. However, grammars are trained separately and can be trained solely on potential word orders. For example, in speech, a grammar may be trained on the articles from the Wall Street Journal from the past three years, while speech is only available for a small fraction of these articles. Thus, grammars might provide additional constraints on the data and simplify recognition.

A common mistake in neural net and HMM training is to provide too little training data. How many examples are enough? This, of course, is highly dependent on the task. In general, the data should be representative of the task domain. Training samples should include as much variance as is possible and reasonable for the task. Another issue is the role of context. While the scope of this thesis includes forty signs, trisine contexts may expand this number to 64,000 three sign combinations. In practice, the constraints of the language curtail this exponentiation. In fact, given the sentence structure used for the task proposed, only 2580 distinct trisines and 364 bisines are possible. The largest class of trisines, "pronoun verb noun," has 1080 members. The largest class of bisines, "verb noun," has 180 members. Note that while context is a powerful tool, it is not necessary for each context to be present in the training data for that context to appear as a result from recognition. Training can be "pooled" from small contexts or even the individual units to gain evidence for a new context during recognition. Of course, the best situation is for all contexts to be seen before recognition, but this is not always possible when a recognition task includes models with a low probability of occurrence.

Weighing these factors against the time and expense of collecting data, 500 sentences were determined sufficient for an attempt at the task. Separating the database into 400 training and 100 test sentences gives an average of between 20 to 80 training examples of each sign (depending on class) with a good likelihood that each bisine occurs at least once.

Chapter 5

Experimentation

The results of the handtracking discussed in the last section can be seen in Figure 5-1. Tracking markers are overlaid on the camera images. The center of the arrow tracks the right hand while the incomplete diamond tracks the left. The length and width of the indicators note the length of the major and minor axes of the bounding ellipses while the angle of the indicators show the angle of the principle axes. Occasionally tracking would be lost (generating error values of 0) due to lighting effects, but recovery was fast enough (within a frame) so that this was not a problem. The 5 frame/sec rate was maintained within a tolerance of a few milliseconds. However, frames where tracking of a hand was lost were deleted. Thus, a constant data rate was not guaranteed.

Of the 500 sentences collected, six had to be thrown out due to subject error or outlier signs. Each sign ranged from approximately 1 to 3 seconds in length. No intentional pauses were given between signs within a sentence, but the sentences themselves were distinct.

Table 5.1: Percentage of words correctly recognized

	test on training	*test on independent test set*
grammar	99.4%	97.0%
no grammar	95.9% (93.5% accuracy)	95.0% (90.7% accuracy)
	(D=13, S=88, I=60, N=2470)	(D=3, S=22, I=21,N=495)

To avoid the boot strapping segmentation process, the evidence in the sentences was evenly distributed between the words. Initial estimates for the means and variances of the output probabilities were provided by iteratively using Viterbi alignment on the training

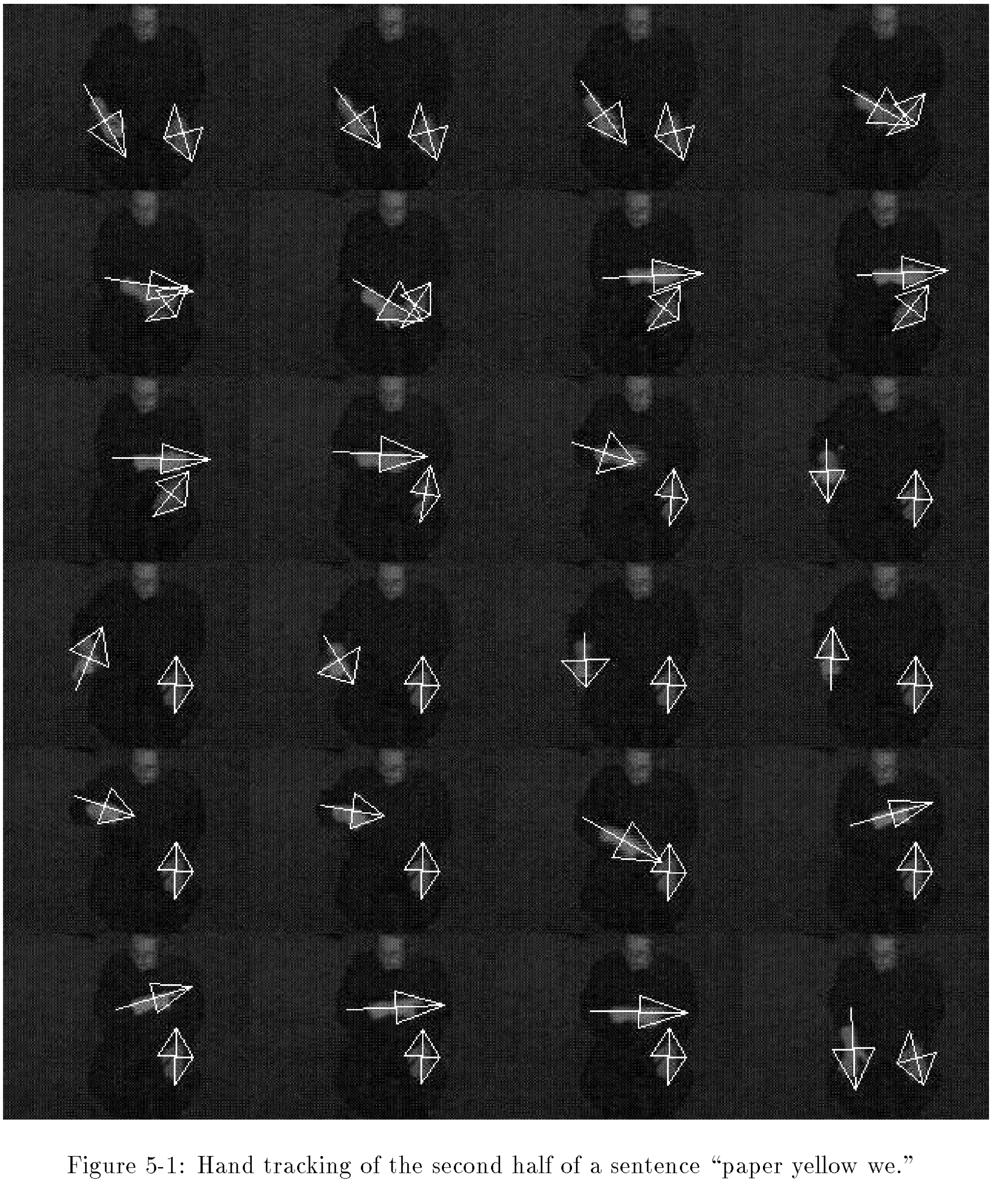

Figure 5-1: Hand tracking of the second half of a sentence "paper yellow we."

data and then recomputing the means and variances by pooling the vectors in each segment. Entropic's Hidden Markov Model ToolKit (HTK) was used as a basis for this step and all other HMM modeling and training tasks. The results from the initial alignment program are fed into a Baum-Welch re-estimator, whose estimates are, in turn, refined in embedded training which ignores any initial segmentation. For recognition, HTK's Viterbi recognizer was used both with and without a strong grammar based on the known form of the sentences. Gesture recognition occurs at a rate five times faster than real time. Contexts were not used, since a similar effect could be achieved with the strong grammar given this data set.

Word recognition results are shown in Table 5.1. When testing on training, all 494 sentences were used for both the test and train sets. For the fair test, the sentences were divided into a set of 395 training sentences and a set of 99 independent test sentences. The 99 test sentences were not used for any portion of the training. Given the strong grammar (pronoun, verb, noun, adjective, pronoun), insertion and deletion errors were not possible since the number and class of words allowed is known. Thus, all errors are substitutions when the grammar is used (and accuracy equals percent correct). However, without the grammar the recognizer is allowed to match the observation vectors with any number of the 40 vocabulary words in any order. Thus, deletion (D), insertion (I), and substitution (S) errors are possible. The absolute number of errors of each type are listed in Table 5.1 as well. The accuracy measure is calculated by subtracting the number of insertion errors from the number of correct labels and dividing by the total number of signs. Note that, since all errors are accounted against the accuracy rate, it is possible to get large negative accuracies (and corresponding error rates of over 100%). Most insertion errors occurred at signs with repetitive motion.

Chapter 6

Analysis and Discussion

While these results are far from being sufficient to claim a "working system" for full ASL recognition, they do show that this approach is promising. The high recognition rate on the training data indicates that the HMM topologies are sound and that the models are converging. Even so, the remaining 6.5% error rate (error rates will be based on accuracy measures) on the "no grammar" case indicates that some fine tuning on the feature set and model is in order. The 3.0% error rate on the independent test set shows that the models are generalizing well. However, a close look at the text produced by the recognition process shows some of the limitations of the feature set. Since the raw positions of the hands were used, the system was trained to expect certain gestures in certain locations. When this varied due to subject seating position or arm placement, the system could become confused. A simple fix to this problem would be to use position deltas in the feature vector instead.

Examining the errors made when no grammar was used shows the importance of finger position information. Signs like "pack," "car," and "gray" have very similar motions. In fact, the main difference between "pack" and "car" is that the fingers are pointed down for the former and clenched in the latter. Since this information was not available to the model, confusion could occur. While recovering general and/or specific finger position may be difficult in real time in the current testing area, simple palm orientation could be used for discrimination instead. In the current system, a simple implementation would be to paint the back of the gloves a different color than the palm.

A more interesting problem in the no grammar results was that signs with repetitive or long gestures were often inserted twice for each actual occurrence. In fact, insertions caused almost as many errors as substitutions. Thus, a sign "shoes" might be recognized as "shoes shoes," which is a viable hypothesis without a language model. However, both problems can be addressed using context training or a statistical or rule-based grammar.

Using context modeling as described before may improve recognition accuracy. While the rule-based grammar explicitly constrained the word order, statistical context modeling would have a similar effect while leaving open the possibility of different sentence structures. In addition, bisine and trisine contexts would help fine-tune the training on the phrase level. However, trisine modeling would not support the tying of the beginning pronoun to the ending pronoun as the grammar does. If task oriented or domain centered sentences were used instead of randomly generated sentences, context modeling and a statistical grammar would improve performance considerably. For example, the random sentence construction allowed "they like pill gray they" which would have a low probability of occurrence in everyday conversation. As such, context modeling would tend to suppress this sentence in recognition unless strong evidence was given for it. In speech and handwriting, a factor of 2 and factor of 4 cut in error rate can be expected for application of contexts and grammars respectively [36]. While ASL does not have the same hierarchy of components as speech and handwriting (letters to words to sentences), the factor of 3 decrease in error when the grammar was applied hints at similar performance increases at the sentence level.

While extending this recognition system to the full 6000 word ASL lexicon would present unforeseen problems, some basic improvements could be made to begin adapting the system to the task:

- Use deltas instead of absolute positions. An alternative is to determine some feature on the subject from which the positions can be measured (for example, the centroid of the subject).
- Add finger and palm tracking information. Exact position information may not be necessary. Some simple starting features may be how many fingers are visible along the contour of the hand and whether the palm is facing up or down.
- Collect appropriate domain or task oriented data and perform context modeling.

These improvements do not address the subject independence issue. Just as in speech, making a system which can understand different subjects with their own variations of the language involves collecting data from many subjects. Until such a system is tried, it is hard to estimate the number of subjects and the amount of data that would comprise a suitable training database. In general, the training database should span the space of required input, with many examples of each context. Independent recognition often places new requirements on the feature set as well. While the modifications mentioned above may be sufficient initially, the development process is highly empirical.

So far, finger spelling has been ignored. However, incorporating finger spelling into the recognition system is a very interesting problem. Of course, changing the feature vector to address finger information is vital to the problem, but adjusting the context modeling is also of importance. With finger spelling, a closer parallel can be made to speech recognition. Here, trisine context is at a lower level than grammar modeling and will have more of an effect. A point of inquiry would be switching between the different modes of communication. Can trisine context be used across finger spelling and signing? Is it beneficial to switch to a separate mode for finger spelling recognition? Can natural language techniques be applied, and if so, can they also be used to address the spatial positioning issues in ASL? The answers to these questions may be key in creating an unconstrained sign language recognition system.

Chapter 7

Summary and Future Work

An unencumbered way of recognizing a subset of American Sign Language (ASL) through the use of a video camera has been shown. Using hidden Markov models, low error rates were achieved on both the training data (.6%) and an independent test set (3%). Instead of invoking complex models of the hands, simple position and bounding ellipse tracking at 5 frames/sec were used to generate the requisite feature vectors. These feature vectors were then converted to text by the recognition process at five times faster than real time.

The recognition system presented does not purport to be a solution to machine ASL recognition. Issues such as finger spelling and the spatial positioning aspects of ASL were ignored. However, immediate extensions of the system toward this goal have been presented. In fact, a constrained, person dependent, complete lexicon system may be possible using the principles described here. Unfortunately, collecting a suitable training database is prohibitive for the 6000 word task.

An immediate future goal is to train the system on a native ASL signer. Now that a stable training process has been created, collecting this data will be simpler, and the experimental results will provide an initial benchmark for the utility of the system.

Research into removing the constraint of using colored gloves for tracking is underway. While a primitive system has already been created, tracking the hands in front of the face and across the variable background of clothing has proven difficult with the current low resolution, single camera environment. Faster hardware, better lighting and calibration, and more sophisticated algorithms are being applied to create a system which can provide not only the current level of tracking, but also an improved, more stable set of features for recognition.

While the current system's components each run in parallel in real time, a direct connection between the two modules has not been established. With this addition, immediate sentence translation should be possible. Since recognition time increases with the log of the size of the lexicon, considerable expansion should be possible while still providing immediacy.

Currently the system ignores semantic context given by facial expressions while signing. By adding expression tracking techniques demonstrated by Essa and Darrell [11], this information might be recovered in the current system. An active camera to provide higher resolution "focus of attention" images might also be added to the current apparatus. This would help alleviate the constraint of constant position when signing as well as provide better data for the tracking software.

Bibliography

[1] Adaptive Optics Associates. *Multi-Trax User Manual.* Adaptive Optics Associates, Cambridge, MA, 1993.

[2] L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Stat.*, 41:164–171, 1970.

[4] Richard A. Bolt and Edward Herranz. Two-handed gesture in multi-modal natural dialog. In *Proceedings of UIST '92, Fifth Annual Symposium on User Interface Software and Technology*, Monterey, CA, 1992.

[5] L. Campbell. Recognizing classical ballet setps using phase space constraints. Master's thesis, Masschusetts Institute of Technology, 1994.

[6] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture, and spoken intonation for multiple conversational agents. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 413–420, July 1994.

[7] C. Charayaphan and A.E. Marble. Image processing system for interpreting motion in American Sign Language. *Journal of Biomedical Engineering*, 14:419–425, September 1992.

[8] T.J. Darrell and A.P. Pentland. Space-time gestures. *IEEE Conf. on Computer Vision and Pattern Rec.*, pages 335–340, 1993.

[9] Trevor Darrell and Alex Pentland. Attention-driven expression and gesture analysis in an interactive environment. MIT Media Lab Perceptual Computing Group Technical Report No. 312, Massachusetts Institute of Technology, 1994.

[10] B. Dorner. Hand shape identification and tracking for sign language interpretation. In *IJCAI Workshop on Looking at People*, 1993.

[11] Irfan Essa, Trevor Darrell, and Alex Pentland. Tracking facial motion. Technical Report 272, MIT Media Lab Vision and Modeling Group, 20 Ames St, Cambridge MA, 1994. To appear in IEEE Workshop on Nonrigid and articulated Motion, Austin TX, Nov 94.

[12] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Technical Report 94-03, Mitsubishi Electric Research Labs., 201 Broadway, Cambridge, MA 02139, 1994.

[13] Yang He and Amlan Kundu. Planar shape classification using hidden markov models. In *Proc. 1991 IEEE Conf. on Computer Vision and Pattern Rec.*, pages 10–15. IEEE Press, 1991.

[14] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, Feb 1983.

[15] Berthold Horn. *Robot Vision.* MIT Press, New York, 1986.

[16] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, 1990.

[17] Tom Humphries, Carol Padden, and Terrence J. O'Rourke. *A Basic Course in American Sign Language.* T. J. Publishers, Inc., Silver Spring, MD, 1980.

[18] B. H. Juang. Maximum likelihood estimation for mixture multivariate observations of markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[19] Myron W. Krueger. *Artificial Reality II.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1991.

[20] F. Kubala, A. Anastasakos, J. Makhoul, L. Nguyen, R. Schwartz, and G. Zavaliagkos. Comparative experiments on large vocabulary speech recognition. In *ICASSP 94*, 1994.

[21] A. Kundu, Y. He, and P. Bahl. Handwritten word recognition: a hidden markov model based approach. volume 22, pages 283–297, 1989.

[22] L. R. Liporace. Maximum likelihood estimation for multivariate observations of markov sources. *IEEE Trans. Information Theory*, IT-28:729–734, 1982.

[23] Pattie Maes, Trevor Darrell, Bruce Blumberg, and Alex Pentland. The ALIVE system: full-body interaction with animated autonomous agents. MIT Media Lab Perceptual Computing Group Technical Report No. 257, Massachusetts Institute of Technology, 1994.

[24] H. Morita, S. Hashimoto, and S. Ohteru. A computer music system that follows a human conductor. *Computer*, 24(7):44–53, July 1991.

[25] Eadweard Muybridge. *Human and Animal Locomotion*, volume 1-2. Dover Publications, Inc., Mineola, N.Y., 1979.

[26] R. Nag, K. H. Wong, and F. Fallside. Script recognition using hidden markov models. In *ICASSP 86*, 1986.

[27] S. Niyogi and E. Adelson. Analyzing gait with spatiotemporal surfaces. Technical Report 290, MIT Media Lab Vision and Modeling Group, 20 Ames St, Cambridge MA, 1994. To appear in IEEE Workshop on Nonrigid and articulated Motion, Austin TX, Nov 94.

[28] D. H. Parish, G. Sperling, and M.S. Landy. Intelligent temporal subsampling of american sign language using event boundaries. *Journal of Experimental Psychology: Human Perception and Performance*, 16(2):282–294, 1990.

[29] H. Poizner, U. Bellugi, and V. Lutes-Driscoll. Perception of american sign language in dynamic point-light displays. volume 7, pages 430–440, 1981.

[30] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, January 1996.

[31] J. M. Rehg and T. Kanade. DigitEyes: vision-based human hand tracking. School of Computer Science Technical Report CMU-CS-93-220, Carnegie Mellon University, December 1993.

[32] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, Jan 1994.

[33] Jennifer Schlenzig, Edd Hunter, and Ramesh Jain. Recursive identification of gesture inputers using hidden markov models. In *Proc. of the Second Annual Conference on Applications of Computer Vision*, pages 187–194, 1994.

[34] Jeffrey Mark Siskind and Quaid Morris. A maximum-likelihood approach to visual event perception. manuscript, 1995.

[35] G. Sperling, M. Landy, Y. Cohen, and M. Pavel. Intelligible encoding of ASL image sequences at extremely low information rates. *Computer Vision, Graphics, and Image Processing*, 31:335–391, 1985.

[36] Thad Starner, John Makhoul, Richard Schwartz, and George Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP 94*, 1994.

[37] W. C. Stokoe, D. C. Casterline, and C. G. Groneberg. *A Dictionary of American Sign Language on Linguistic Principles.* Linstok Press, London, 1976.

[38] Y. Suenaga, K. Mase, M. Fukumoto, and Y. Watanabe. Human reader: an advanced man-machine interface based on human images and speech. *Systems and Computers in Japan*, 24(2):88–101, 1993.

[39] T. Takahashi and F. Kishino. Hand gesture coding based on experiments using a hand gesture interface device. *SIGCHI Bulletin*, 23(2):67–73, 1991.

[40] Shinichi Tamura and Shingo Kawasaki. Recognition of sign language motion images. volume 21, pages 343–353, 1988.

[41] V. C. Tartter and K. C. Knowlton. Perceiving sign language from an array of 27 moving spots. volume 289, pages 676–678, 1981.

[42] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequetial images using hidden markov model. In *Proc. 1992 IEEE Conf. on Computer Vision and Pattern Rec.*, pages 379–385. IEEE Press, 1992.

[43] S. J. Young. *HTK: Hidden Markov Model Toolkit V1.5.* Cambridge University Engineering Department Speech Group and Entropic Research Laboratories Inc., Washington DC, December 1993.