Parallel Hidden Markov Models for American Sign Language Recognition

Christian Vogler and Dimitris Metaxas
Department of Computer and Information Science, University of Pennsylvania
Philadelphia, PA 19104-6389
cvogler@gradient.cis.upenn.edu, dnm@central.cis.upenn.edu

Abstract

The major challenge that faces American Sign Language (ASL) recognition now is to develop methods that will scale well with increasing vocabulary size. Unlike in spoken languages, phonemes can occur simultaneously in ASL. The number of possible combinations of phonemes after *enforcing linguistic constraints is approximately* 5.5×10^8. *Gesture recognition, which is less constrained than ASL recognition, suffers from the same problem.*

Thus, it is not feasible to train conventional hidden Markov models (HMMs) for large-scale ASL applications. Factorial HMMs and coupled HMMs are two extensions to HMMs that explicitly attempt to model several processes occuring in parallel. Unfortunately, they still require consideration of the combinations at training time.

In this paper we present a novel approach to ASL recognition that aspires to being a solution to the scalability problems. It is based on parallel HMMs (PaHMMs), which model the parallel processes independently. Thus, they can also be trained independently, and do not require consideration of the different combinations at training time.

We develop the recognition algorithm for PaHMMs and show that it runs in time polynomial in the number of states, and in time linear in the number of parallel processes. We run several experiments with a 22 sign vocabulary and demonstrate that PaHMMs can improve the robustness of HMM-based recognition even on a small scale. Thus, PaHMMs are a very promising general recognition scheme with applications in both gesture and ASL recognition.

1. Introduction

Let us imagine a computer that could interact with users via speech and gestures. Now, let us imagine another computer that could interact with users only via a commonplace point-and click windowed interface. Which one would gain higher acceptance?

Obviously, speech and gesture would provide the most natural means for humans to interact with a computer. Whereas speech recognition has made significant advances in the past decade, gesture recognition has been lagging behind. Yet, gesture is an integral part of everyday communication among humans [9], so it is likely to be important in human-computer interaction as well. A working gesture and speech recognition system would probably entail a major paradigm shift away from point-and click-type user interfaces.

There is the danger that such a paradigm shift would leave behind deaf people, who cannot use speech to communicate. To accommodate them, it is necessary to substitute sign language for speech. Sign language recognition research can also greatly benefit gesture recognition research, because sign language has structure. This structure makes it easier to develop and test methods on sign language recognition first before applying them to gesture recognition.

The major challenge that faces American Sign Language (ASL) recognition research now is how to make recognition approaches scale with large vocabularies. There are two critical aspects to scalability in ASL recognition: The first aspect consists of breaking down the signs into their constituent phonemes. This breakdown is necessary, because there is a small, limited number of phonemes in ASL as opposed to the number of signs, particularly when the signs are inflected. The second aspect consists of modeling these phonemes. Modeling the phonemes properly is more difficult than in speech recognition, because speech is largely sequential, whereas many phonemes in ASL occur in parallel during the course of a sign. For example, in many signs both the left and right hands move in a large number of possible combinations (see Figure 5 for an example). Attempting to capture all the possible different combinations of phonemes statically — for example, by training a hidden Markov model (HMM) for each combination — would be futile for anything but the smallest vocabularies.

In this paper, we present a new approach to the parallel modeling of the phonemes within an HMM framework. In previous work, researchers have proposed extensions to HMMs to model the interaction of several processes in parallel, such as factorial hidden Markov models (FHMMs) [3], or coupled hidden Markov models (CHMMs) [2]. These extensions require modeling the interactions of the processes during the training phase, and thus require training examples of every conceivable combination of actions that can occur in parallel. Thus, it is doubtful that FHMMs and CHMMs will scale well in ASL recognition.

We present parallel hidden Markov models (PaHMMs) as an alternative extension to HMMs, in an attempt to overcome the scalability problems of FHMMs and CHMMs. They are based on modeling the different processes as independent processes that can be trained separately. There is linguistic evidence that ASL can be modeled at least partially as independent processes [8]. Hence, PaHMMs stand a much better chance of being scalable than FHMMs and CHMMs. Because gesture recognition is even less constrained than ASL recognition, PaHMMs are highly significant to gesture recognition research, as well.

In the following sections, we first give an overview of related work in ASL recognition. Then we briefly describe the theory of regular HMMs and its extension to FHMMs and CHMMs. We show why scalability is a concern and then present PaHMMs as a possible solution, along with a new recognition algorithm. In the experimental section, we show how PaHMMs can be applied to continuous ASL recognition with a 22 sign vocabulary, where the signs have been broken down into their constituent phonemes, and how PaHMMs perform compared to regular HMMs.

2. Related Work

Recently, research has begun to shift from isolated sign language recognition toward continuous recognition. T. Starner and A. Pentland use a view-based approach with a single camera to extract two-dimensional features as input to HMMs with a 40-sign vocabulary and a strongly constrained sentence structure consisting of a pronoun, verb, noun, adjective, and pronoun in sequence [11].

We used HMMs to model phonological aspects of ASL [14, 13] with an unconstrained sentence structure. We used the Movement-Hold phonological model by S. Liddell and R. Johnson [8] as the basis for our work. In this paper we extend our earlier work.

R. H. Liang and M. Ouhyoung use HMMs for continuous recognition of Taiwanese Sign Language with a vocabulary between 71 and 250 signs [7]. Their work is based on Stokoe's model [12], which, however, has been superseded by other phonological models, such as [8]. Unlike other work in this area, they perform explicit temporal segmentation, and they integrate the handshapes, positions, orientations, and movements of signs at a higher level than the HMMs.

H. Hienz, B. Bauer, and K.-F. Kraiss use HMMs for continuous video-based sign language recognition of German Sign Language with a 52-sign vocabulary [5]. In addition, they use unigram and bigram language models to improve recognition performance.

3. Hidden Markov Models

Before we discuss different extensions to HMMs and related work, we give an overview on the aspects of HMM theory relevant to this paper.

3.1. Overview on HMMs

Hidden Markov models are a type of statistical model embedded in a Bayesian framework. In their simplest form, an HMM λ consists of a set of N states $S_1, S_2, \ldots, S_N$. At regularly spaced discrete time intervals, the system transitions from state S_i to state S_j with probability a_{ij}. The probability of the system initially starting in state S_i is π_i.

Each state S_i generates output $O \in \Omega$, which is distributed according to a probability distribution function $b_i(O) = P\{\text{Output is } O | \text{System is in } S_i\}$. In most recognition applications $b_i(O)$ is actually a mixture of Gaussian densities.

We now describe the main algorithm used for continuous recognition. For a discussion of how to estimate (i.e., train) the parameters of an HMM and how to compute the probability that an HMM generated an output sequence, see [10].

In many continuous recognition applications, the HMMs corresponding to individual signs are chained together into a network of HMMs. Then the recognition problem is reduced to finding the most likely state sequence through the network. That is, we would like to find a state sequence $Q = Q_1, \ldots, Q_T$ over an output sequence $O = O_1, \ldots, O_T$ of T frames, such that $P(Q, O|\lambda)$ is maximized. Using

$$\delta_t(i) = \max_{Q_1, \cdots, Q_{t-1}} P(Q_1 Q_2 \ldots Q_t = S_i, O|\lambda), \qquad (1)$$

and by induction

$$\delta_{t+1}(i) = b_i(O_{t+1}) \cdot \max_{1 \le j \le N} \{\delta_t(j) a_{ji}\}, \qquad (2)$$

$$P(Q, O|\lambda) = \max_{1 \le i \le N} \{\delta_T(i)\}, \qquad (3)$$

the Viterbi algorithm computes this state sequence in $O(N^2 T)$ time, where N is the number of states in the HMM network. Note that the Viterbi algorithm implicitly segments the observation into parts as it computes the path through the network of chained HMMs.

In this paper, we adapt a different formulation of the recognition algorithm, called the **token passing algorithm** [15], for ASL recognition. It works as follows:

- Each state S_i contains at time t a **token** denoting $\delta_t(i)$ from Equation 1.
- At time $t+1$, for each S_i, pass tokens $\text{tok}_{ij}(t+1) = \delta_i(t) a_{ij}$ to all states S_j connected to S_i.
- Finally, for each state S_j, pick $\max_i\{\text{tok}_{ij}(t+1)\}$, and update this token to denote $\delta_j(t+1) = \text{tok}_{ij}(t+1) b_j(O_{t+1})$.

The token passing algorithm is equivalent to the Viterbi algorithm. The main difference between the two algorithms is that the former updates the probabilities via the outgoing transitions of a state, whereas the latter updates the probabilities via the ingoing transitions of a state. Hence, only the order, in which the probabilities are updated, is different. Token passing has the advantage of making the concept of a path through the network explicit, because each token can carry additional path information.

3.2. Extensions to HMMs

Regular HMMs are capable of modeling only one process evolving over time. Sign language, however, consists of parallel, possibly interacting, processes[1]. For example, the left and the right hands move in different ways during the course of a sign, depending on whether the sign is one-handed or two-handed. Modeling these parallel processes with a single HMM would require that the processes evolve in lockstep, passing through the same state at the same time. This lockstep property of regular HMMs is unsuitable for many applications. For example, if a one-handed sign precedes a two-handed sign, the weak hand often moves to the location required by the two-handed sign before the strong hand starts to perform it. Thus, it is necessary to extend the HMM framework, so as to model parallel processes in a satisfactory manner.

In past research, two fundamentally different methods of extending HMMs have been described. The first method models the C processes in C separate HMMs, effectively creating a meta-state in an C-dimensional state space. It combines the output of the C HMMs in a single output signal, such that the output probabilities depend on the C-dimensional meta-state (Figure 1). Such models are called factorial hidden Markov models (FHMMs). Because the output probabilities depend on the meta-state, an optimal training method based on expectation maximization would take time exponential in C. Z. Ghahramani and M. Jordan describe approximate polynomial-time training methods based on mean-field theory [3].

The second method consists of modeling the C processes in C HMMs, whose state probabilities influence one another, and whose outputs are separate signals. That is, the transition from state $S_t^{(i)}$ to $S_{t+1}^{(i)}$ in the HMM for process i does not only depend on the state $S_t^{(i)}$, but on the $S_t^{(j)}$ states in all processes, where $1 \leq j \leq C$ (Figure 2). Such HMMs are called coupled hidden Markov models (CHMMs). M. Brand, N. Oliver, and A. Pentland describe polynomial-time training methods and demonstrate the advantages of CHMMs over regular HMMs in [2].

Unfortunately, modeling the interaction between parallel processes solves only part of the problem of modeling the phonemes in ASL. We also need to solve the problem

[1]How these processes interact is still under debate in sign language linguistics.

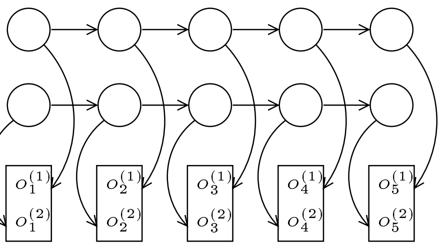

Figure 1. FHMMs: The output is combined. $O_i^{(n)}$ denotes the output of the nth process at the ith frame.

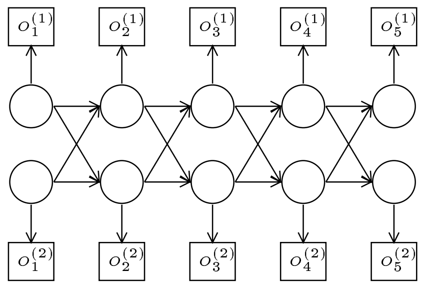

Figure 2. CHMMs: The output is separate, but the states influence one another.

of the sheer number of possible combinations of parallel phonemes in ASL, as explained in the next section.

4. The Challenge: Scalability

Just like spoken languages, ASL can be broken down into a limited set of phonemes that form the language's basic building blocks. Although this set of phonemes is larger than in spoken languages, it is still limited to approximately 100 total, comprising approximately 30 handshapes, 8 hand orientations, 20 major body locations, and 40 movements [12, 8].

Unlike in speech, phonemes occur both sequentially and simultaneously in ASL, giving way to a bewildering number of possible combinations. If we consider both hands, there are $(30 \times 8 \times 20 \times 40)^2 \approx 10^{10}$ possible combinations. In reality, there are constraints on the handshape and movement of the weak hand during two-handed signs, but even then the number of possible combinations would still be approximately 5.5×10^8.

Because the output probabilities of FHMMs depend on the meta-state, and because transitions in CHMMs depend on the states of all the coupled processes, it would be necessary to model the possible combinations separately. But training an order of 10^8 HMMs is not practical, even with tying the parameters of similar HMMs. Likewise, even if we assume that only pairs of processes, such as the handshape and movement of a hand, need to be coupled, the number of combinations is still too large.

Looking at ASL from the whole-sign level does not help either. Even though the cataloged vocabulary of ASL consists of only approximately 6000 signs, many signs can be highly inflected. Verbs like "GIVE" can be modified in the starting location, ending location, handshape, and type of movement, so as to indicate subject, recipient, object, and manner of action. Thus, the number of possible cases to consider on the whole-sign level would be several orders of magnitude larger than 6000.

In an attempt to overcome these problems with scalability, we now investigate PaHMMs as a new paradigm of coupling HMMs for sign language recognition.

5. A New Approach: Parallel HMMs

PaHMMs model the C processes with C independent HMMs with separate output (Figure 3). Unlike CHMMs, the state probabilities influence one another only within the same channel. That is, PaHMMs are essentially regular HMMs that are used in parallel.

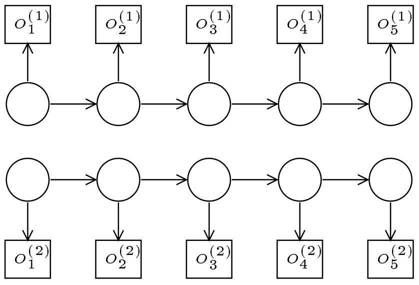

Figure 3. PaHMMs: The output is separate, and the states of separate processes are independent. $O_i^{(n)}$ denotes the output of the nth process at the ith frame.

H. Hermansky, S. Tibrewala, and M. Pavel, as well as H. Bourlard and S. Dupont, first suggested the use of PaHMMs in the speech recognition field [4, 1]. They break down the speech signal into subbands, which they model independently, so as to be able to exclude noisy or corrupted subbands, and merge the subbands during recognition with multi-layered perceptrons. They demonstrate that subband modeling can improve recognition rates. Note that the goal of subband modeling differs from our goal of making ASL recognition methods scale. Subband modeling is concerned with eliminating unreliable parts of the speech signal, whereas we would like to develop a tractable method of considering all parts of the ASL signal.

PaHMMs are based on the assumption that the separate processes evolve independently from one another with independent output. The justification for using this independence assumption is that there is linguistic evidence that the different processes of ASL can be viewed as acting with a high degree of independence on the phoneme level [8]. Our experiments in Section 6 show that this assumption is indeed justified.

As a consequence, the HMMs for the separate processes can be trained wholly independently. Thus, the problem of modeling all possible combinations of phonemes disappears. It is necessary to consider only an order of $(30 + 8 + 20 + 40) \times 2$ HMMs instead of an order of 10^8 HMMs (see the previous section for an explanation of the numbers).

5.1. Combination of the Processes

At some stage in the recognition process, it is necessary to merge the information from the HMMs representing the C different channels. We would like to find (in log probability form)

$$\max_{Q^{(1)},\dots,Q^{(C)}} \{\log P(Q^{(1)},\dots,Q^{(C)}, O^{(1)},\dots,O^{(C)}|\lambda_1,\dots,\lambda_C)\}, \tag{4}$$

where $Q^{(i)}$ is the state sequence of channel i with output sequence $O^{(i)}$ through the HMM network λ_i. Furthermore, the $Q^{(i)}$ are subject to the constraint that they all follow the same sequence of signs. Because we assume the channels to be independent, the merged information consists of the product of the probabilities of the individual channels, so we can rewrite (4) as

$$\begin{aligned}&\max_{Q^{(1)},\dots,Q^{(C)}} \{\log P(Q^{(1)},\dots,Q^{(C)},\\ &O^{(1)},\dots,O^{(C)}|\lambda_1,\dots,\lambda_C)\} =\\ &\max_{Q^{(1)},\dots,Q^{(C)}} \left\{\sum_{i=1}^{C} \log P(Q^{(i)},O^{(i)}|\lambda_i)\right\}.\end{aligned} \tag{5}$$

Because HMMs assume that successive outputs are independent, we rewrite (5) as

$$\begin{aligned}&\max_{Q^{(1)},\dots,Q^{(C)}} \left\{\sum_{i=1}^{C} \log P(Q^{(i)},O^{(i)}|\lambda_i)\right\} =\\ &\max_{Q^{(1)},\dots,Q^{(C)}} \left\{\sum_{j=1}^{W}\sum_{i=1}^{C} \log P(Q^{(i)}_{(j)},O^{(i)}_{(j)}|\lambda_i)\right\},\end{aligned} \tag{6}$$

where we split the output sequences into W segments, and $Q^{(i)}_{(j)}$ and $O^{(i)}_{(j)}$ are the respective state and observation sequences in channel i corresponding to segment j. Intuitively, this equation tells us that we can combine the probabilities as many times as desired at any stage of the recognition process, including the whole-sign level or the phoneme level.

It is desirable to weight the channels on a per-word basis, because in some two-handed signs the weak hand does not move. Such signs could be easily confused with one-handed signs where the weak hand happens to be in a position similar to the one required by the two-handed sign. In these situations, the strong hand should carry more weight than the weak hand. If we let $\omega_j^{(i)}$ be the weight of word j in channel i, the desired quantity to maximize becomes (from Equation 6)

$$\max_{Q^{(1)},\dots,Q^{(C)}} \left\{ \sum_{j=1}^{W} \sum_{i=1}^{C} \omega_j^{(i)} \log P(Q_{(j)}^{(i)}, O_{(j)}^{(i)} | \lambda_i) \right\}, \quad (7)$$

where $\sum_i \omega_j^{(i)} = C$ for fixed j.

Before we describe how the token passing algorithm described in Section 3 needs to be modified for PaHMMs, we need to consider a subtle point. Consider using two channels to model the movements of the strong and the weak hands in ASL. What does the weak hand do in a one-handed sign? From a recognition point of view, we do not care, and thus we should assign a probability of one to anything that the weak hand does during the course of a one-handed sign. Unfortunately, doing so would bias recognition toward one-handed signs, because the average log probabilities for one-handed signs would then be twice as large as the average log probabilities for two hands. Instead, we define the probability of the weak hand to be the same as the probability of the strong hand for one-handed signs.

5.2. The Recognition Algorithm

In principle, adapting the token passing algorithm to PaHMMs consists of applying the regular token passing algorithm to the HMMs in the separate channels, and combining the probabilities of the channels at word or phoneme ends according to (7). See Figure 4 for an example with two channels (e.g., left and right hands).

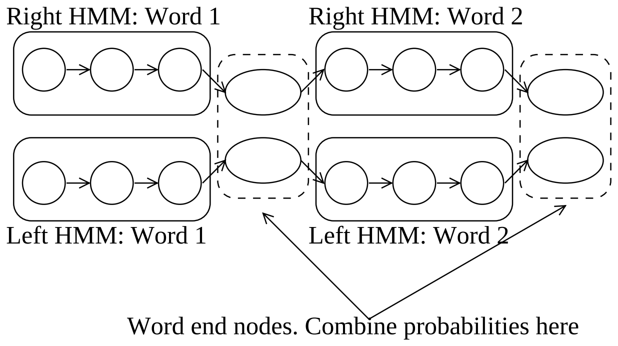

Figure 4. The tokens are passed independently in the HMMs for the left and the right hands, and combined in the word end nodes.

In practice, the recognition algorithm is more complicated, because it must enforce the constraint that the paths $Q^{(i)}$ all touch exactly the same sequence of words. It does not make sense to combine the probabilities of tokens from different paths. The easiest way to enforce this constraint is to assign unique path identifiers to the tokens at each word end and to combine the probabilities of only those tokens that have the same path identifier.

In addition, a path in channel k that contributes to maximizing (7) does not necessarily maximize the marginal probability $\sum_{j=1}^{W} \log P(Q_{(j)}^{(k)}, O_{(j)}^{(k)} | \lambda_k)$. To overcome the potential discrepancy between maximizing the joint and marginal probabilities, each state needs to keep track of a set of the first few best tokens, each with a unique path identifier.

To ensure that the algorithm assigns the probabilities of the strong hand to the weak hand when it encounters a one-handed sign, we define two operations:

- **Join**($node$) takes the tokens of the weak hand in word end node $node$ and attaches them to the tokens of the strong hand in the same word end node. The attached token must have the same path identifier as the token that it is attached to.
- **Split**($node$) detaches the weak hand tokens from the strong hand tokens in word start node $node$. It checks for each detached token, whether the last sign in the path was one-handed or two-handed. If it was one-handed, **Split** updates the probabilities of the detached tokens with the probabilities of the strong hand for the last sign. Then it merges the tokens with the existing tokens of the weak hand in the same word start node.

If we denote the number of output frames with T, the modified token passing algorithm is given in Algorithm 1.

Algorithm 1 Token passing algorithm for PaHMMs

```
1: Initialize the tokens in the start nodes of the HMM network with $\log p = 0$.
2: for $t = 1$ to $T$ do
3:   for $c = 1$ to $C$ do
4:     for each $state$ in all HMM states do
5:       Pass the tokens in $state$ to the adjacent states and merge them with the tokens in the adjacent states.
6:     end for
7:   end for
8:   for $c = 1$ to $C$ do
9:     for each $node$ that is a word end node do
10:      Combine the token probabilities.
11:      if $node$ is a two-handed sign then
12:        Join($node$).
13:      end if
14:      for each $node'$ adjacent to $node$ do
15:        Pass the tokens in $node$ to $node'$
16:        if $node'$ is a two-handed sign then
17:          Split($node'$).
18:        end if
19:      end for
20:    end for
21:  end for
22: end for
```

If we assume that the token sets in each state have cardinality M and are stored as lists sorted by log likelihood, passing the token set from one single state to another takes $O(M)$ time. Hence, step 5 takes $O(NM)$ time, where N is the number of states in the HMM network. Combining the token probabilities in step 10 and **Join** take $O(M \log M)$ time each. **Split** takes $O(M)$ time, because it is essentially a token set merge. Steps 14–15 take $O(NM)$ time.

Because $N \gg M$, we can deduce that the entire algorithm runs in $O(T(CN^2M + CN(M \log M + NM))) = O(N^2T\ CM)$ time. That is, it takes time linear in the number of channels and in the number of tokens per state.

6. Experiments

We ran several continuous recognition experiments with 3D data to test the feasibility of modeling the movements of the left and the right hands with PaHMMs. Our database consisted of 400 training sentences and 99 test sentences over a vocabulary of 22 signs. We performed all training and testing with a heavily modified version of Entropic's Hidden Markov Model Toolkit (HTK).

We collected the sentences with an Ascension Technologies MotionStar™ 3D tracking system for speed reasons, but they could have just as well been collected with a vision-based tracking system, such as [6] (see also Figure 5). The sentence structure was constrained only by what is grammatical in ASL.

Figure 5. These images show the 3D tracking of a complex two-handed sign ("MAIL")

6.1. ASL Modeling

Following Liddell and Johnson's Movement-Hold model [8], we modeled each sign as a series of movement and hold segments. During a movement segment, at least one of the parameters of a sign changes, whereas during a hold segment, all the parameters are stationary. A sign typically consists of three to five segments.

The total number of unique segments was 89 for the right hand and 51 for the left hand, so we trained a total of 140 HMMs. In a testament to the clear advantage of phoneme-based modeling over whole-sign-based modeling, many HMMs had more than 30 training examples available.

We used an 8-dimensional feature vector for each hand. Six features consisted of 3D positions and velocities relative to the base of the signer's spine. For the remaining two features, we computed the largest two eigenvalues of the positions' covariance matrices over a window of 15 frames centered on the current frame. In normalized form, these two eigenvalues provide a useful characterization of the global properties of the signal [13]. Note that our goal is to evaluate a novel recognition algorithm, not the merits of different features.

6.2. Comparison of PaHMMs and Regular HMMs

To establish a baseline, we first ran an experiment using only the 8-dimensional features (3D position, 3D velocities, and eigenvalues of the positions' covariance matrices) of the right hand with regular HMMs. The results are given in Table 1. We did not test FHMMs, CHMMs, or regular HMMs with both hands, because even for the small 22-sign vocabulary the number of occurring phoneme combinations was far too large for the 400-sentence training set. The goal of these experiments was to demonstrate that PaHMMs can outperform regular HMMs while preserving scalability, not to investigate whether PaHMMs perform better or worse than FHMMs and CHMMs.

Level	Accuracy	Details
sentence	80.81%	H = 80, S = 19, N = 99
sign	93.27%	H=294, D=3, S=15, I=3, N=312

Table 1. Regular HMMs: Results of the recognition experiments. 80.81% of the sentences were recognized correctly, and 93.27% of the signs were recognized correctly. H denotes the number of correct sentences or signs, D the number of deletion errors, S the number of substitution errors, I the number of insertion errors, and N the total number of sentences or signs in the test set.

An analysis revealed that there were only seven sentences with incorrectly recognized two-handed signs. Each of these seven sentences involved a single substitution error. Thus, the maximum recognition rate that we could expect from using PaHMMs to model both hands in this experiment was 87.88% on the sentence level and 96.47% on the sign level. Table 2 shows the actual recognition rates with PaHMMs, with merging of the token probabilities at the phoneme level.

Level	Accuracy	Details
sentence	84.85%	H = 84, S = 15, N = 99
sign	94.23%	H=297, D=3, S=12, I=3, N=312

Table 2. PaHMMs: Results of the recognition experiments, with merging of the token probabilities at the phoneme level. See Table 1 for an explanation of the terminology.

Of the seven sentences with two-handed signs that the regular HMMs failed to recognize, the PaHMMs recognized four correctly. One of the other three sentences now contained an additional substitution error in a one-handed sign. All other sentences were not affected. That is, the PaHMMs recognized every single sentence correctly that was already recognized correctly by the regular HMMs.

We view this result as evidence that PaHMMs can improve recognition rates over regular HMMs, with no significant tradeoffs in recognition accuracy. This result also contributes evidence toward validating the assumption that the parallel processes in ASL can be modeled independently.

6.3. Factors Influencing PaHMM Accuracy

There are two factors that can potentially influence the recognition accuracy of PaHMMs. The first factor is the required cardinality M of the token set in each state. Because the time complexity of the recognition algorithm is linear in M, the cardinality should be as small as possible. The second factor is the level of merging the token probabilities. Is it better to perform the merging at the phoneme level or at the whole-sign level?

Table 3 shows the results for token set cardinalities of 2, 3, 5, and 8. Recognition accuracy does not seem to be affected by cardinalities beyond 3. The log probabilities of the tokens are not significantly affected either. We expect that using more than two channels will not have a significant effect on the required cardinality of the token sets, provided that the HMMs in each channel have been well trained.

Card.	Sent. Accuracy	Sign Accuracy
2	82.83%	92.95%
3	84.85%	94.23%
5	84.85%	94.23%
8	84.85%	94.23%

Table 3. Effect of token set cardinality on recognition rates, with merging of the token probabilities at the phoneme level.

Table 4 shows the effect of merging the token probabilities at the whole-sign level. The level of merging has a small effect on recognition rates, but it is not significant.

Merge level	Sent. Accuracy	Sign Accuracy
Sign-level	84.85%	94.23%
Phoneme-level	84.85%	94.55%

Table 4. Effect of the level of token probability merging on recognition rates. In both cases, the token set had a cardinality of 3.

7. Conclusion

We demonstrated that PaHMMs can improve the robustness of ASL recognition even on a small scale. Because PaHMMs are potentially more scalable than other extensions to HMMs, they are an interesting research topic for gesture and sign language recognition. Future research should establish how PaHMMs behave with larger vocabularies, and particularly with highly inflected signs that can exhibit a large number of phoneme combinations within one single sign. Future research should also add hand configuration and orientation, and facial expressions as new channels to the PaHMM framework.

Acknowledgments

This work was supported in part by a NSF Career Award NSF-9624604, ONR Young Investigator Proposal, NSF IRI-97-01803, AFOSR F49620-98-1-0434, and NSF EIA-98-09209.

References

[1] Herve Bourlard and Stephane Dupont. Subband-based speech recognition. *ICASSP*, 1997.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled Hidden Markov Models for complex action recognition. *CVPR*, 1997.

[3] Zoubin Ghahramani and Michael I. Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29:245–275, 1997.

[4] Hynek Hermansky, Sangita Tibrewala, and Misha Pavel. Towards ASR on partically corrupted speech. *ICSLP*, pages 462–465, 1996.

[5] H. Hienz, B. Bauer, and K.-F. Kreiss. HMM-based continuous sign language recognition using stochastic grammars. *Gesture Workshop*, Gif sur Yvette, France, 1999.

[6] I. Kakadiaris, D. Metaxas, and R. Bajcsy. Model based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. *CVPR*, pages 81–87, 1996.

[7] R.-H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. *Third International Conference on Automatic Face and Gesture Recognition*, pages 558–565, Nara, Japan, 1998.

[8] Scott K. Liddell and Robert E. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, 64:195–277, 1989.

[9] David McNeill. *Hand and mind: what gestures reveal about thought*. University of Chicago Press, Chicago, 1992.

[10] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[11] Thad Starner and Alex Pentland. Visual recognition of American Sign Language using Hidden Markov Models. *International Workshop on Automatic Face and Gesture Recognition*, pages 189–194, Zürich, Switzerland, 1995.

[12] William C. Stokoe. *Sign Language Structure: An Outline of the Visual Communication System of the American Deaf*. Studies in Linguistics: Occasional Papers 8. Linstok Press, Silver Spring, MD, 1960. Revised 1978.

[13] Christian Vogler and Dimitris Metaxas. Toward scalability in ASL recognition: Breaking down signs into phonemes. *Gesture Workshop*, Gif sur Yvette, France, 1999.

[14] Christian Vogler and Dimitris Metaxas. Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. *SMC*, 1997.

[15] S. Young, N. Russell, and J. Thornton. Token passing: a conceptual model for connected speech recognition systems. Technical report, F_INFENG/TR38 Cambridge University, 1989.