Hidden Markov Models for Greek Sign Language Recognition

VASSILIA N. PASHALOUDI, KONSTANTINOS G. MARGARITIS
Parallel Distributed Processing Laboratory
Department of Applied Informatics
University of Macedonia
156 Egnatia str., P.O. Box 1591, 54006, Thessaloniki,
GREECE

Abstract: - Sign languages are the basic means of communication between hearing impaired people. A translator is usually needed when a deaf person wants to communicate with persons that do not speak sign language. The work presented in this paper aims at developing a system that could recognize and translate isolated and continuous Greek Sign Language (GSL) sentences. Input is obtained by using a feature extracting method from two-dimensional images. The feature vectors produced by the method are then presented as input to Hidden Markov Models (HMMs) for recognition. We use a vocabulary of 26 greek words (nouns, pronouns, adjectives, verbs) and achieve a recognition rate of 98% for isolated recognition and 85.7% for continuous recognition.

Key-Words: - Hidden Markov Models, sign language, sign language recognition, gesture recognition.

1 Introduction

Sign language (SL) is the usual method of communication for hearing-impaired people, who often have to communicate with other people through a sign-language interpreter. However, they cannot always communicate when they wish, since the number of interpreters is limited. Therefore, there is an increasing need for an automatic SL translation system.

The term "sign" in SL most often represents a whole word, unless it is differently specilied by some researchers in their work. Signs can be performed by one or two hands and they are called one-handed or two-handed respectively. In one-handed signs the hand that performs the sign is always the same and is called *dominant* hand. A sign in sign language is a movement of the dominant hand or of both hands. However, in most sign languages special words, such as first and last names, have to be spelled out, by using sequences of the static morphs, that represent the alphabet letters. Thus, one aspect of SL recognition is the recognition of hand morphs, called *postures*, and the other aspect is the recognition of signs perfomed by hand movements, called *gestures*. Both aspects are significant, though the latter is more complicated, since it compensates with the recognition of hand morphs that change through time. Furthermore, the recognition of SL sentences can be confronted in two ways: *isolated recognition*, where each sign has start and endpoint, and *continuous recognition*, where there are no marked boundaries between signs. In the latter case the recognition performance of each sign is affected by the preceding and the subsequent sign (*co-articulation*).

Different methods have been used for sign language recognition, involving Neural Networks, Image Processing Algorithms, Fuzzy Systems ([1]-[6]) or Hidden Markov Models. Until recently many systems use "datagloves" as input devices for the recognition of SL gestures. These approaches limit the user's freedom of movement. Video-based techniques are less intrusive and therefore more comfortable to utilize.

In the following sections we give a summary of the theory of Hidden Markov Models and discuss their use in Sign Language (SL) recognition, mentioning also related work. Finally we outline data collection and provide experimentation results for isolated and continuous recognition.

2 Hidden Markov Models

Hidden Markov Models (HMMs) are a type of statistical model. They have been successfully used in speech recognition and recently in handwriting, gesture and sign language recognition. We now give a summary of the basic theory of HMMs. Detailed theory is presented in [7].

Given a set of N states s_i we assume that a system can be in one of these states at each time interval; we can describe the transitions from one state to another at each time step t as a stochastic process. The transition probability to reach state s_i in the first time step is denoted as π_i. Assuming that the transition probability α_{ij} of state s_i to state s_j only depends on the preceding states, we call this process a Markov chain. The further assumption, that the actual transition only depends on a very preceding state leads to a first order Markov chain. We can now define a second stochastic process that produces, at each time step t, symbol vectors x. The emission probability of a vector x only depends on the actual state, but not on the way the state was reached. The emission probability density $b_i(x)$ for vector x at state s_i can either be discrete or continuous.

This doubly stochastic process is called a Hidden Markov Model (HMM), if only the vectors x are observable, but not the state sequence. A HMM, λ, is defined by its parameters $\lambda = (A,B,\pi)$. The NxN matrix A represents the transition probabilities a_{ij} from state s_i to s_j, B denotes the vector of the emission densities $b_i(x)$ of each state s_i and π is the vector of the initial transition probabilities π_i.

The first problem we have to solve, when working with HMMs, is *the evaluation problem:* Given the Observation sequence O, compute the probability $P(O|\lambda)$, that a HMM $\lambda=(\pi,A,B)$ generated O. This problem corresponds to maximum likelihood recognition of an unknown data sequence with a set of HMMs, each of which corresponds to a sign. For each HMM, the probability, that it generated the unknown sequence, is computed, and then the HMM with the highest probability is selected as the recognized sign. For computing this probability a method called **Forward-Backward** algorithm is used [7].

Another problem we have to cope with is that, given the parameters of a HMM λ and an observation sequence O of vectors O_t of the signal, we have to find the state sequence q, that emits, with a high probability, the same symbol vectors as observed from the signal. This problem can be solved with the **Viterbi** algorithm [7]. This is called the *decoding problem*.

The last problem we have to deal with is the *estimation problem:* Adjust the parameters of an HMM λ such that they maximize $P(O|\lambda)$ for some O. This problem corresponds to training the HMMs with data, such that they are able to recognize previously unseen data correctly after training. No analytical calculation method is known for maximizing $P(O|\lambda)$ for given observation sequences, but an iterative procedure, called the **Baum-Welch** procedure, maximizes $P(O|\lambda)$ locally [7].

The most commonly used HMM *topology* is the **left-right** model or else **Bakis** model, where transitions only flow forward from lower states to the same state or higher states.

3 HMMs in Sign Language Recognition

3.1 General Remarks

HMMs are an attractive choice for processing both speech and two-dimensional sign data, because their state-based nature enables them to capture variations in the duration of signs, by remaining in the same state for several time frames.

We shall first discuss some issues of speech recognition, which is the first area where HMMs have been widely used. The specific issues apply to SL recognition, too.

The recognition process can be examined from two different aspects: according to the kind of element it attempts to recognize (*phoneme* or *word*) and according to whether there are artificial pauses between signs or not (*isolated* or *continuous* recognition).

Word-based and Phoneme-based recognition. According to the first aspect, there are **whole-word** or *word-based* systems, where separate HMMs are trained for each word, and *phoneme-based* systems, where separate HMMs are trained for each phoneme. A **phoneme** is defined to be the smallest contrastive unit in a language; that is a unit that distinguishes one word from another. In a *phoneme-based* recognition system, for each word in the dictionary the pronounciation must be specified as a sequence of these phonemes. In addition, adding a new word to the vocabulary involves only modification to the pnonouncing dictionary and task grammar. In a *word-based* recognition system adding a new word would additionally require the creation of a new HMM and also the retraining of all the HMM models over the whole set of training examples.

The main advantage in the phoneme-based recognition is that the number of phonemes is limited in any language, including SL, wherever they have been itemized, as opposed to the unlimited number of words that can be built from phonemes. Thus, for large-scale applications the most effective and commonly used method is the phoneme-based recognition, whilst for small-scale

applications **whole-word** training is more appropriate.

Isolated and Continuous recognition. Examining the recognition process from another point of view, we can distinguish it to *isolated* and *continuous* recognition.

Isolated Recognition assumes that there are clearly marked boundaries between words. Such a boundary could be silence. Each HMM is firstly trained for the vocabulary word it represents, using a number of examples of the word. The recognition process extracts the signal corresponding to each word individually. It then picks the HMM that yields the maximum likelihood for that signal as the recognized word. Training the HMMs to maximize recognition performance is also comparatively straightforward. Initially, all signals in the training set are labeled. For each word in the dictionary, the training procedure computes the mean and covariance matrix over the data available for that word and assigns them uniformly as the initial output probabilities to all states in the corresponding HMM. It also assigns initial transition probabilities, uniformly to the HMM's states. The training procedure then runs the **Viterbi** algorithm repeatedly on the training samples, so as to align the training data along the HMM's states. The aligned data are then used to estimate better output probabilities for each state individually. This alignment yields major improvements in recognition performance, because it increases the chances of the **Baum-Welch reestimation** algorithm converging to an optimal or near-optimal maximum. After constructing the HMMs, the training procedure finishes by reestimating each HMM in turn with the Baum-Welch reestimation algorithm, which maximizes $P(O|\lambda)$ locally.

In *Continuous Recognition* on the other hand, there is no silence between segments of speech, so the straightforward method of using silence to distinguish speech segments fails. HMMs offer the compelling advantage of being able to segment the streams of speech automatically with the Viterbi algorithm.

There is a training method for continuous recognition, called *embedded-training* [21], that is used to overcome the lack of large amount of data, which is very important in achieving good model estimates in continuous recognition. *Embedded-training* uses the **Baum-Welch** procedure as for the isolated case, but, rather than training each model individually, all models are trained in parallel. The method involves connecting HMMs together in sequence. Each model in the sequence corresponds to the assumed underlying symbol: either whole words or sub-words (such as phonemes). Each iteration of this procedure concatenates the HMMs corresponding to the individual symbols in a training sentence into a single large HMM. It then reestimates the parameters of the large HMM with a single iteration of the Baum-Welch algorithm. The reestimated parameters, however are not immediately applied to the individual HMMs. Instead, they are pooled in accumulators, and applied to the individual HMMs only after the training procedure has iterated over all sentences in the training set.

Word-based and Phoneme-based Recognition in SL Sign language recognition can also be confronted in both above mentioned aspects. Considering SL in the first point of view, we should make clear the concept of **phoneme** in sign languages: the **phoneme** could be the "**morph**" of the hand, that is a posture among a specific predefined set of postures, such as those appearing in Liang-Ouhyoung model [10]; or it could be considered as a **movement** or a **hold**, according to Liddell and Johnson [19]. The last approach is what Vogler and Metaxas used in [17] and [18] in order to break words-signs of American Sign Language into phonemes.

Isolated and Continuous Recognition in SL In parallel to speech recognition, there is another way in approaching the recognition problem, according to whether there are artificial pauses between signs or not. *Isolated* recognition attempts to recognize a single sign at a time. Hence, it is based on the assumption that each sign can be individually extracted and recognized. *Continuous* recognition, on the other hand, attempts to recognize an entire stream of signs, without any artificial pauses or marked boundaries between signs.

3.2 Review of Previous Work

Nam and **Wohn** [8] use an HMM-based method for recognizing the space-time hand movement pattern. They use "datagloves" and achieve up to 80% accuracy. **Yamato** *et al.* [9] use HMMs to recognize image sequences of 6 tennis strokes, which are captured by video. They use a small feature vector and achieve up to 90% accuracy. **Grobel** and **Assam** [15] use HMMs to recognize isolated signs with 91.5% accuracy out of 262 sign vocabulary. They extract the features from video recordings of signers wearing coloured gloves. **Liang** and **Ouhyoung** [10], [11] have worked on Taiwanese SL recognition. They use "datagloves" to obtain data and a **language model** to recognize sentences.

In the field of continuous recognition **Starner**, **Pentland** [12], and **Weaver** [13], [14] use a view-based approach to extract two-dimentional features as input to HMMs with a 40 word vocabulary. They use cotton gloves or bare hands in their experiments and attain accuracies from 92% up to 99%. **Vogler** and **Metaxas** [16], [17], [18], have worked on American SL recognition. They use video cameras to capture signs and context-dependent HMMs to improve recognition results on a 53 sign vocabulary. **Hienz, Bauer** and **Kraiss** [20] have worked on German SL recognition, using video-based techniques and stochastic language model to improve performance. They achieve recognition rates of 95%, working on a 52 sign vocabulary.

3.3 Our approach

In GSL, there is no widely accepted library of morphs. The only morphs, that have been encountered and itemized without dispute, are the morphs that represent the 24 alphabet letters. Those morphs can be used in sequences to form first names or specialized words that are not common in GSL. For the rest of the words in GSL each word is formed by moving one or both hands and can be considered as an image sequence through time. As an immediate consequence of the lack of morph-itemization, we have chosen the option of whole-word HMMs in our approach. An HMM for each one of the words is constructed.

Each HMM must be trained over a number of examples of the word it represents, so that in the recognition phase it could give the maximum probability among all the other HMMs, for the word it represents.

At this point we should make clear the way we use the word “sign” in our work. A sign in SL usually represents a word, as has been already mentioned. Though, the way we use the term in our work, is to describe a sequence of postures, each one representing an alphabet letter of GSL. This sequence forms the word when we “spell” it. Thus, in our approach we use image sequences of the GSL alphabet letters for all the words, which naturally have different lengths, to form a sign. The same method could be used in recognizing image sequences that have been captured at sequential time intervals, through the formation of a gesture, representing a word, instead of GSL alphabet letters.

4 Feature Extraction of images and Data Preparation

4.1 Feature Extraction

In our approach we use monochrome bitmap images of the GSL alphabet letters, with size 100x100 pixels, as the initial data. Each bitmap image is transformed to a 2-dimensional array of 1's and 0's, placed in the corresponding positions of “black” and “white” pixels, respectively. Thus, we can easily extract a feature vector from each image, whose elements consist of the numbers of “black” pixels in each scan line. Therefore, for an image of 100x100 we extract a feature vector of length 100. Feature vectors are then transformed to HTK format.

4.2 Data Collection and Preparation

We used the 26 word vocabulary listed in Table 1

Pronouns	I, you, he, we, you (plural), they
nouns	car, train, airplain, dog, cat, house, horse, bicycle
verbs	want, don'twant, have, don'thave, love, hate
Adjectives	black, white, red, yellow, small, big

Table 1 The complete 26 word vocabulary

The goal in choosing the vocabulary was to be able to express sentences that could have occured in a natural conversation.

We have built a large database of all the letters, transformed in size or location or rotation or combination of these attributes. To ensure rotation, location and handsize invariance, we form the training examples for each word by retrieving randomly letter-morphs, from the above described database. In that way, we enhance systems' tolerance to these factors' changes.

Isolated Recognition For the isolated recognition case, we use training files, each one of which holds a single example of a word, with no leading or trailing silence. Each sign has at least 5 examples available for the training set, and 2 examples available for the test set. Thus a total of 172 examples has been collected over a range of 26 signs.

Each one of these files is formed by the feature vectors extracted from the images representing the letters of the word. The length of the words ranges between 3-7 image-letters. Image-letters are randomly retrieved from the previously described database; this means that, not all the letters have the same orientation of the hand, nor are they posed in the same location of the image area.

Continuous Recognition The same policy is followed in the continuous SL recognition case, where we have to form sentences. The training examples in this case are sequences of feature vectors that stand for the sequences of the words constituing a sentence.

We have collected up to now 134 continous GSL sentences, each between 2 and 5 signs long, with a total of 512 signs. Each sign was between 3-7 frames long, as in the isolated recognition case. The only constraints on the order and occurence of signs were those dictated by the syntax of GSL.

5 Experiments

We performed isolated and continuous GSL recognition experiments by using Entropic's Hidden Markov Model Toolkit (HTK) Version 3.0 for all HMM modelling and training tasks.

The number of states and the topology used for the HMMs is important. Sign language as a time-varying process lends itself naturally to a left-right model topology, which we also used here. The initial topology for an HMM can be determined by estimating how many different states are involved in specifying a sign. Finding the optimum number of states, which depends on the frame rate and the complexity of the signs involved, is an empirical process. While better results might be obtained by tailoring different topologies for each sign, a four states HMM with skip transitions was determined to be sufficient for modelling all signs. A skip transition allows a model to emulate a 3 or 4 state HMM, depending on the training data for the particular sign.

The output probabilities are single Gaussian densities with diagonal covariance matrices.

For recognition, HTK's Viterbi recognizer is used with a **grammar**, according to GSL syntax:

[pronoun] noun {adjective} verb

where square brackets [ ] denote optional selection of an item and braces { } denote zero or more repetitions of an item. Thus, we do not use a **strong** grammar, which would reduce the error rates, in order to have a syntax closer to the natural way of "speaking" with GSL.

Comparing Isolated and Continuous recognition From the analysis performed by the HTK tool **Hresults,** comes up that isolated recognition gives better results than continuous. More specifically, word accuracy in the isolated recognition case reaches 98% for the independent test set, while in the continuous recognition case it gets up to 96.7%.

The analysis of the experiments for the isolated and continuous recognition cases are presented in Table 2 and Table 3 respectively.

	Corr %	Acc %	
SENT	98.08		H=51,S=1, N=52
WORD	98.08	98.08	H=51, D=0, S=1, I=0, N=52

Table 2 Isolated Recognition Results

	Corr %	Acc %	
SENT	85.71		H=54, S=9, N=63
WORD	96.70	92.92	H=205, D=0, S=7, I=8, N=212

Table 3 Continuous Recognition Results

Since continuous recognition is under developement, we used a reduced number of the collected sentences for the moment.

In the tables presented above, the sentence-level accuracy is based on the total number of files which are identical to the transcription files. The word accuracy is based on Dynamic Programming matches between the label files and the transcriptions. H is the number of correct labels, D is the number of deletions, S is the number of substitutions, I is the number of insertions and N is the total number of labels in the defining transcription files. The percentage number of labels correctly recognized is given by

$$Correct = \frac{H}{N} x100\%$$

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.

$$Correct = \frac{H - I}{N} x100\%$$

7 Conclusions and Future Work

It should be clear that the present work is a first approach to GSL recognition. We have worked on isolated and continuous GSL recognition. Through the use of Hidden Markov Models, low error rates were achieved on both the training set and an independent test set, without invoking complex models of the hand.

Much more work should be done in the field of continuous recognition. Since a great number of GSL sentences is needed in order to train HMMs for continuous SL, our next goal becomes the collection of the required number of sentences. The

134 sentences that we collected up to now and used as training sentences were proved insufficient and gave low recognition rates. The method of **embedded training** has also been used in training, but as it claims a large number of training utterances, too, it gave poor results up to now. We expect that, with a larger training set and context modelling, lower error rates will come out.

We also intend to expand the vocabulary and be able to form sentences from a wide variety of words.

References:

[1] C. Charayaphan, A. Marble, Image Processing System for Interpreting Motion in American Sign Language, *Journal of Biomedical Engineering*, Vol.14, 1992, pp. 419-425.

[2] S. Fels, G. Hinton, Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer, *IEEE Transactions on Neural Networks*, Vol.4, No.1, 1993.

[3] M. Waldron, S. Kim, Isolated ASL Sign Recognition System for Deaf Persons, *IEEE Transactions on Rehabilitation Engineering*, Vol.3, No.3, 1995.

[4] E. Holden, R. Owens, G. Roy, Hand Movement Classification using an Adaptive Fuzzy Expert System, *International Journal of Expert Systems Research and Applications*, Vol.9, Iss.4, 1996, pp. 465-480.

[5] M. Su, A fuzzy Rule-Based Approach to Spatio-Temporal Hand Gesture Recognition, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol.30, No.2, 2000, pp. 276-281.

[6] V. Pashaloudi, K. Margaritis, On Greek Sign Language Alphabet Character Recognition: using Back-Propagation Neural Networks, *Proccedings of the 5th Hellenic European Research on Computer Mathematics and its Applications (HERCMA) Conference*, 2001.

[7] L.R. Rabiner, A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of IEEE*, Vol.77, No.2, 1989, pp. 257-286.

[8] Y. Nam, K.Y. Wohn, Recognition of Space-Time Hand-Gestures using HMM, *ACM Symbosium on Virtual Reality Software and Technology*, Hong Kong, 1996, pp. 51-58.

[9] J. Yamato, J. Ohya, K. Ishii, Recognizing Human Action in Time-Sequential Images using Hidden Markov Models, *Proceedings of the International Conference on Computer Vision IEEE Press*, 1992, pp. 379-385.

[10] R.H. Liang, M. Ouhyoung, A Sign Language Recognition System using Hidden Markov Models and Context Sensitive Search, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* HongKong, 1996, pp. 59-66.

[11] R.H. Liang, M. Ouhyoung, A Real-time Continuous Gesture Recognition System for Sign Language, *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition* Nara Japan, 1998, pp. 558-565.

[12] T. Starner, A. Pentland, Real-time American Sign Language Recognition from Video using Hidden Markov Models, *Perceptual Computing Section Technical Report No.375,* MIT Media Lab Cambridge MA, 1996.

[13] T. Starner, J. Weaver, A. Pentland, A Wearable Computer Based American Sign Language Recognizer, *The First International Symposium on Wearable Computers*, 1997, pp. 130-137.

[14] T. Starner, J. Weaver, A. Pentland, Real-Time American Sign Language Recognition using Desk and Wearable Computer Based Video, *Pattern Analysis and Machine Intelligence*, 1998.

[15] M. Assan, K. Grobel, Video-Based Sign Language Recognition using Hidden Markov Models, *Proceedings of the International Gesture Workshop*, Bielefeld Germany, 1997, pp.97-109.

[16] C. Vogler, D. Metaxas, ASL Recognition based on a Coupling Between HMMs and 3D Motion Analysis, *Proceedings of the International Conference on Computer Vision* Mumbai India, 1998, pp. 363-369.

[17] C. Vogler, D. Metaxas, Towards Scalability in ASL Recognition: Breaking Down Signs into Phonemes, *Gesture Workshop*. Gif sur Yvette, France, 1999.

[18] C. Vogler, D. Metaxas, Parallel Hidden Markov Models for American Sign Language Recognition, *Gesture Workshop*, Corfu Greece, 1999

[19] S. Liddell, R. Johnson, *American Sign Language: The phonological base*. Sign Language Studies 64, 1989, pp. 195-277.

[20] H. Hienz, B. Bauer, K.F. Kraiss, HMM-Based Continuous Sign Language Recognition using Stochastic Grammars, *Gesture Workshop* Gif sur Yvette, France, 1999.

[21] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, Woodland, Ph. *The HTK Book*.