Isolated Sign Language Recognition using Hidden Markov Models

Kirsti Grobel and Marcell Assan

Lehrstuhl für Technische Informatik, RWTH Aachen
Ahornstraße 55, D-52074 Aachen, Germany
E-mail: {grobel, assan}@techinfo.rwth-aachen.de

ABSTRACT

This paper is concerned with the video-based recognition of isolated signs. Concentrating on the manual parameters of sign language, the system aims for the signer dependent recognition of 262 different signs. For Hidden Markov Modelling a sign is considered a doubly stochastic process, represented by an unobservable state sequence. The observations emitted by the states are regarded as feature vectors, that are extracted from video frames. The system achieves recognition rates up to 94 %.

1 INTRODUCTION

During the recent years human-machine-interfaces have experienced a growing interest. Systems for the analysis of body motion have been developed carefully as a first step for gesture interaction between the user and the computer. A special case of body motion is gestures, i.e. motion of hands and arms. Gestures can express a certain meaning - such as pointing gestures or command gestures - and support the verbal communication. If a community of deaf people has assigned defined meanings to certain gestures, they are called signs, being part of a sign language.

Many sign languages, like Sign Language of the Netherlands, are characterised by manual and non-manual parameters [2]. Manual parameters are handshape, orientation, location and movement of hands. Non-manual parameters are line of sight, facial expression, poise, etc. If two signs only differ in one parameter they are called a minimal pair. Signs can be divided into one-handed and two-handed signs. For one-handed signs the action of only one hand is required, where a person generally takes the same hand, known as the dominant hand. For two-handed signs, the other hand, called the non-dominant hand, performs the sign together with the dominant hand.

By now many systems use datagloves as input devices for the recognition of gestures or sign language, e.g. [8]. These approaches suffer from limiting the user's freedom of movement. Video-based techniques are less intrusive and therefore more comfortable to utilise. The developed system should only use a single colour camera in order to minimise the necessary hardware components and the resulting adjustments.

The following problems, resulting from the above requirements, must be taken into account:

- Signs vary in time and space. Even if a person tries to perform the same sign twice, slight changes of speed and position of the hands will occur.
- As the system uses only a single camera, the 3D-space is projected on a 2D-plane, resulting in loss of depth information.
- While signing some fingers can be occluded, as they are hidden behind other parts of the hands or arms.
- The position of the test person in front of the camera can vary. A shift of the signer in one direction and rotation around his body axis must be considered.

The property of Hidden Markov Models (HMMs) to compensate time and amplitude variances of signals has been proven for speech and character recognition [9]. This property makes HMMs appear an ideal approach for sign language recognition. Like speech, sign language can be considered as a nondeterministic time signal, not as a word or phoneme sequence but as a sequence of signs. Unlike speech recognition, where the smallest unit is the phoneme, linguists have not agreed on first proposals on sub-units for signs [3]. Thus we model each sign with one HMM. For both, training and recognition, feature vectors must be extracted from each video frame and then inputted to the HMM.

The presented system for the recognition of isolated signs only regards the manual parameters, as non-manual parameters often have grammatical functions or emphasise emotions. The system is capable of recognising 262 different signs of the Sign Language of the Netherlands. The aim is signer dependent recognition, i.e. the same person trains and tests the system.

This paper is structured as follows. Section 2 gives an overview of related work in this field. Section 3 mentions important details of the theory of HMM and section 4 explains how we adapted this theory to sign language recognition. A description of the experiments and their results can be found in section 5 and in the last section we summarise the main issues discussed in this paper.

2 RELATED WORK

Different methods have been used for gesture recognition. Cui and Weng [5] recognise 26 hand gestures with a recognition rate of 93%. They analyse the shoulder-chest area and segment the handshape from the image. Therefore regions-of-interest in different sizes are compared with trained patterns. The final decision about the gesture is made using a nearest neighbour algorithm.

Yamato et al. [13] presented the first HMM approach recognising six tennis strokes with a rate of about 90% using a 25x25 pixel sub-sampled video image as the feature vector.

An unusual approach is reported from Schlenzig et al. [10]. They use a single universal HMM and a finite state estimator for the determination of gestures. Hand poses are classified with a neural net. Sequences of gestures of six static hand poses are used to control a robot vehicle, yielding classification rates of 96%.

0-7803-4053-1/97/$10.00 © 1997 IEEE

Campbell et al. [4] investigate the rotation and shift invariance of ten different feature vectors. They concentrate on the trajectories of both hands only. The test set of 54 sentences built from 18 Tai Chi gestures yields recognition accuracies of 95%, using a stereo camera.

Starner and Pentland [12] present a system for the recognition of short sentences in American Sign Language, with a vocabulary of 40 signs. Signs are modelled with four-states-HMM. They use a single camera and plain cotton gloves in two colours or no gloves for a second test. They achieve recognition accuracies between 75% and 99%, allowing only the simplest syntactical structures.

Though the results of related work are promising, all systems concentrate on very reduced vocabulary sizes with a maximum of 50 hand poses, gestures or signs. Heading for sign language recognition these sizes are completely insufficient. To get closer to this goal our work investigates a vocabulary of 262 signs and seeks solutions for the problems arising from the greater number of minimal pairs.

3 THEORY OF HIDDEN MARKOV MODELS

Given a set of N states s_i we can describe the transitions from state to state 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 a_{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 the 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 $b_i(\mathbf{x})$ 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 $\lambda = (\pi, \mathbf{A}, \mathbf{B})$. π stands for the vector of the initial transition probabilities π_i, the N×N matrix **A** represents the transition probabilities a_{ij} from state s_i to s_j and finally, **B** denotes the vector of the emission densities $b_i(\mathbf{x})$ of each state s_i.

Having defined the HMM we have to cope with the problem, that given the parameters of a HMM λ and an observation sequence O of vectors O_t of the signal, how to determine the state sequence, that best models the signal. In other words, how to find the state sequence, that emits, with a high probability, the same symbol vectors as observed from the signal. This problem can be solved with the Viterbi algorithm:

1. Initialisation: $1 \le i \le N$

$$\delta_1(i) = \pi_i b_i(O_1)$$
$$\psi_1(i) = 0$$

2. Recursion: $2 \le t \le T;\ 1 \le j \le N$

$$\delta_t(j) = \max_i \{\delta_{t-1}(i) a_{ij}\} b_j(O_t)$$
$$\psi_t(j) = \arg\max_i \{\delta_{t-1}(i) a_{ij}\}$$

3. Termination:

$$P^*(O|\lambda) = \max_i \{\delta_T(i)\}$$
$$q_T^* = \arg\max_i \{\delta_T(i)\}$$

4. Backtracking the state sequence:

$$q_T^* = \psi_{t+1}(q_{t+1}^*)$$

In these formulas i and j are running indices for the states and T the number of time steps t. For the first t observations, $\delta_t(j)$ is the highest probability along the path, that ends in state s_j. $\psi_t(j)$ stores for every state s_j at each time step t the best predecessor state. At the end the algorithm outputs the state sequence **q***. Furthermore it produces an approximation $P^*(O|\lambda)$ for the probability of the observation sequence O, given the HMM λ. Though $P^*(O|\lambda)$ is the maximum probability of all paths rather than the sum of the probabilities of all paths, the Viterbi algorithm was often successfully implemented for the decoding task [9].

Another problem is the estimation of the parameters of a HMM λ, given one or more observation sequences O. Though no analytical calculation method is known to date, the Viterbi training represents a solution, that iteratively adjusts the parameters π, **A** and **B**. In every iteration, the most likely path through a HMM is calculated. This path gives the new assignment of observation vectors O_t to the states s_j. The steps in this procedure for discrete emission densities are [11]:

1. Choose a start model $\lambda^{(0)}$.
2. For n=1,2,... until end condition is met
   - 2.1 Decode the state sequence **q*** using the Viterbi algorithm, given $\lambda^{(n)}$
   - 2.2 Update the parameters with **q***

$$\hat{\pi}_i = \frac{\chi[q_1^* = s_i]}{\sum_{i=1}^{N} \chi[q_1^* = s_i]}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \chi[q_t^* = s_i, q_{t+1}^* = s_j]}{\sum_{j=1}^{N} \sum_{t=1}^{T-1} \chi[q_t^* = s_i, q_{t+1}^* = s_j]}$$

$$\hat{b}_j(\mathbf{x}_k) = \frac{\sum_{t=1}^{T-1} \chi[q_t^* = s_j, O_t = \mathbf{x}_k]}{\sum_{k=1}^{K} \sum_{t=1}^{T-1} \chi[q_t^* = s_j, O_t = \mathbf{x}_k]}$$

   - 2.3 Set $\lambda^{(n)} = (\hat{\pi}, \hat{\mathbf{A}}, \hat{\mathbf{B}})$

In the update equations χ is an abbreviation for $\chi[\text{true}]=1$ and $\chi[\text{false}]=0$. K denotes the number of different vectors of the emission alphabet of the states. An alternative algorithm is the Baum-Welch reestimation procedure. It leads to improved parameters at every iteration step, while the Viterbi training only guarantees that the new parameters are at least as good as in the last step. Still the two procedures show practically the same results, where the Baum-Welch reestimation requires more calculation steps [9].

To avoid quantisation errors and to achieve a simple representation consisting of few parameters, continuous emission probabilities can be used. One example is the M-component Laplacian mixture density

$$b_j(\mathbf{x}) = \sum_{m=1}^{M} b_{jm}(\mathbf{x}) = \sum_{m=1}^{M} c_{jm} g_{jm}(\mathbf{x}, \mu_{jm}, \sigma_{jm})$$

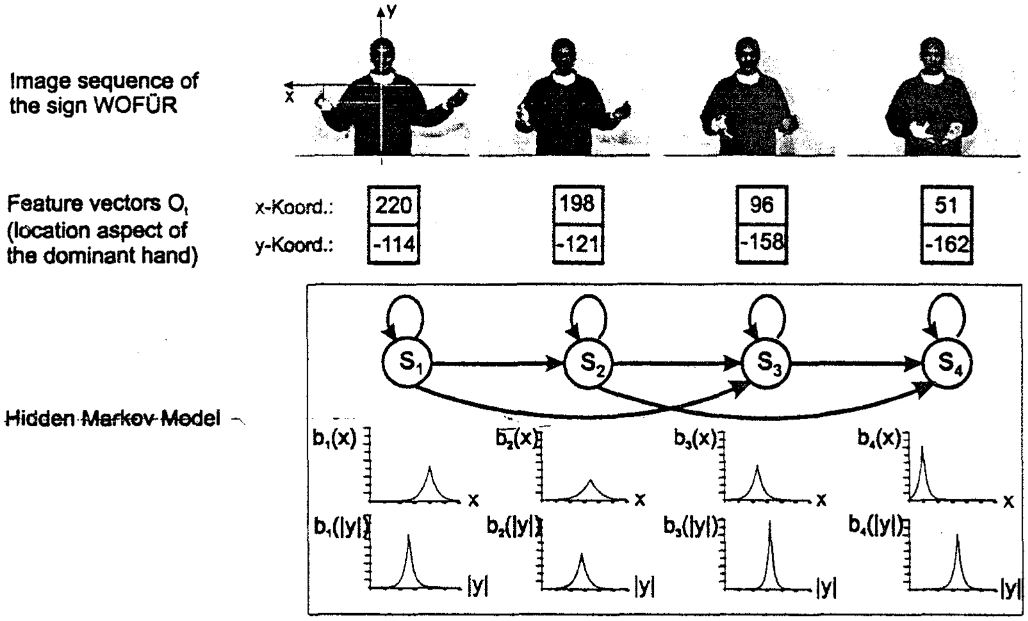

Fig. 1: Modelling the sign "WOFÜR" with a 4-state-Bakis-HMM.

where c_{jm} is the mixture coefficient, μ_{jm} the mean vector and σ_{jm} the vector of the absolute deviation of the m-th mixture component g_{jm} of state s_j. For continuous emission densities the equations of the Viterbi training procedure has to be modified, such that c_{jm}, μ_{jm} and σ_{jm} are updated at each iteration using the observation vectors assigned by q^* to state s_j. The Viterbi training is also called segmental k-means algorithm as for continuous densities, the algorithm starts with one mixture component per state and in the following steps the procedure successively splits a component in two if a likelihood criterion is met.

4 Hidden Markov Models for Sign Language Recognition

Having discussed the theory of HMMs the question arises, how one can find the observation and state sequence in sign language. Fig. 1 illustrates the modelling of a sign with a HMM. The upper line shows four images of the sign "WOFÜR" as observed by the video camera. For each image an observation vector is extracted, which is - for simplicity - displayed only with two features, representing the position of the right hand. Assuming that always four different images are recorded for this sign it would be suitable to choose a linear four-state HMM, with transitions only from one state to the next. For the training procedure, several four-state image sequences would be recorded and as the assignment of the feature vectors to the states would be obvious, emission distributions for every feature of every state could be calculated. Dropping the assumption of a constant image number, the signer is allowed to perform the sign slower or faster. Choosing the Bakis topology for the HMM with additional transitions as displayed in Fig. 1 the system can compensate different speed of signing. This model allows transitions to the same state, the next and the one after next, and has been frequently used for speech recognition [9].

Now the assignment of the vectors to the states is no longer trivial, i.e. it is hidden.

4.1 Feature Extraction

As the HMMs require feature vectors, an important step is the determination and extraction of features. In order to allow real-time data acquisition and to easily retrieve information about the performed handshape, the signer wears simple coloured cotton gloves. Taking into account the different amount of information represented by the handshape of the dominant and non-dominant hand and the fact that many signs can be discriminated only by looking at the dominant hand, different gloves have been chosen: one with seven colours - marking each finger, the palm and the back of the dominant hand - and a plain glove in an eighth colour for the non-dominant hand.

A threshold algorithm generates in/out-code for the colours of the gloves, skin, body and background. In the next processing step the size and the centre of gravity (COG) of the colour areas are calculated and a rule-based classifier estimates the position of the shoulders and the central vertical axis of the body silhouette [7]. Using this information we build a feature vector that reflects the manual parameters of sign language, without explicitly modelling them. In other words at the end of the recognition the system outputs the whole sign, but not the characterising parameters.

Tab. 1 shows how the parameters of sign language are represented by the feature vector. Extracting the position of the hands during signing, we obtain the **location** of a sign. Thus the COGs of each hand are taken into account, where the COG for the dominant hand is determined as the mean position of all COGs of its colour areas. To compensate the shift variance of the signer in front of the camera the co-ordinates are related to a fixed point on the body. The x-co-ordinates are calculated relative to the central vertical axis of the body silhoutte, the y-co-ordinates to the height of the right shoulder. As we use only one camera, the z-co-ordinate is neglected.

An optimal representation of the **handshape** of the dominant hand must be shift and rotation invariant. Using the co-

Tab. 1: Representation of the parameters of sign language within the feature vector. The numbers in brackets designate the number of features.

Hands	Parameters	Features
Dominant hand	Location	x-co-ordinate, relative to central vertical body axis (1) y-co-ordinate, relative to the height of the right shoulder (1)
	Handshape	distances of COGs of all colour areas to each other (20)
	Handshape/ Orientation	size of colour areas (7)
	Orientation	angles φ_1 and φ_2 of fingers (2)
Non-dominant hand	Location	x-co-ordinate, relative to central vertical body axis (1) y-co-ordinate, relative to the height of the right shoulder (1)
	Handshape/ Orientation	size of colour area (1)
Both hands	Location	distance of the COGs of both hands (1)

ordinates of the COGs of all fingers relative to a point on the body would achieve shift invariance, but the features would not be invariant to a rotation around the z-axis. Therefore we chose all distances of COGs of the colour areas to each other. By this means the values of these handshape features do not change for a shift in the x- and y-directions or a rotation around the z-axis. These features contain information about the bending of the fingers - e.g. extended, round, flat or sharply bent - and about the relative position of one finger to the others - e.g. far away or close. As not all fingers must be visible at any time, using redundant features can avoid loosing information. Thus we include all possible distances of the colour areas of the dominant hand into the feature vector. Additionally, we use the sizes of the colour areas to describe the handshape. By this means we obtain a measure by which fingers must be visible for a certain handshape.

For the representation of the **orientation** of a hand the rotation around all three axes must be considered. Again using the sizes of the colour areas we yield orientation information as to the sizes of the areas in- or decrease when the hand is rotated around the x- and y-axis. If, as for the non-dominant hand, a plain glove was used, it would be impossible to distinguish between most of the possible orientations. Taking several colour areas into account these ambiguities are diminished for the dominant hand. If the dominant hand is turned around the z-axis, the sizes stay the same. Therefore we introduced another two orientation features. We define a system of co-ordinates with the origin being the mean position of the palm and the back of hand. Within this system an angle to an axis for each of the four fingers can be calculated. This way we build angle φ_1 as the absolute value of the mean of the four finger angles counting from the x-axis and φ_2 as the absolute value of the mean angle counting from the y-axis, both ranging from 0° to 180°. Considering the estimate of the probability distribution, when the mean value is near 180° it is preferable using φ_1 and φ_2 instead of a single angle φ ranging from -180° to 180°. Let the real distribution of φ be a single Laplacian distribution with mean 180°. In this case the segmental k-means training would inaccurately estimate both halfs of the distribution with a separate distribution. Using φ_1 and φ_2 diminishes this effect.

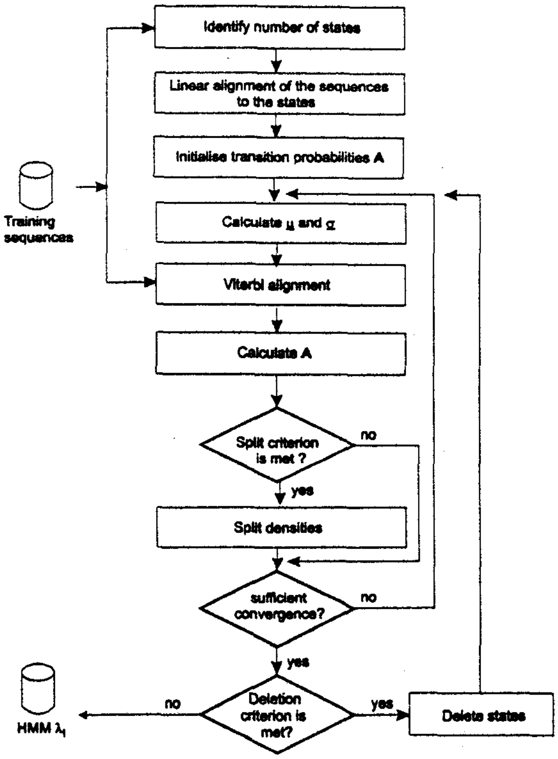
Identify number of states
Linear alignment of the sequences to the states
Initialise transition probabilities A
Training sequences
Calculate $\underline{\mu}$ and $\underline{\sigma}$
Viterbi alignment
Calculate A
Split criterion is met ?
no
yes
Split densities
sufficient convergence?
no
yes
HMM λ_i
no
Deletion criterion is met?
yes
Delete states

Fig. 2: Viterbi training for a HMM

Due to occlusion of fingers, some colour areas can be partly or completely invisible. Thus, distance, position and angle features are only available, if the corresponding colour areas sizes exceed a threshold. The feature vector contains different numbers of features per parameter. To avoid an overemphasis of one parameter, we introduced specific weights for each type of feature.

4.2 Training

Having defined the feature vector HMMs can be trained. The Viterbi training of HMMs for isolated signs is shown in Fig. 2. The first step is the determination of the number of states of the HMM. A fixed number of states for all signs is not suitable, as the database contains very short signs with around four image frames and different, longer signs with about 30 frames. Even the length of one sign can vary considerably. Therefore the number of vectors in the shortest training sequence is chosen as the initial number of states for the HMM of the corresponding sign. Next, the system assigns the vectors of each sequence evenly to the states and initialises the matrix **A**, i.e. all transitions are set equally probable. Using the initial assignment the mean and deviation values of all components of the emission distributions of each state can be calculated. In later iterations, when the algorithm created more than one mixture component per state, in this step the deviation is pooled over all components. The initial HMM $\lambda^{(0)}$ is now complete, as $\pi_1=1$ for a Bakis-HMM.

Given $\lambda^{(0)}$ the Viterbi algorithm determines a new assignment $\mathbf{q}^{(u)*}$ for each training sequence $\mathbf{O}^{(u)}$. With $\mathbf{q}^{(u)*}$, the transition

Tab. 2: Training sets

Training set	Persons	Samples per sign
Training 1	Person 1	10
Training 2	Person 2	5
Training 3	Person 1 + 2	15

Tab. 3: Test sets

Test set	Persons	Samples per sign
Test 1	Person 1	4
Test 2	Person 2	5
Test 3	Person 1 + 2	9

Tab. 4: Recognition results for isolated signs

Vocabulary	Training 1		Training 2		Training 3		
	Test 1	Test 2	Test 1	Test 2	Test 1	Test 2	Test 3
43	98.8 %	91.1 %	86.6 %	95.8 %	98.3 %	100 %	99.2 %
150	93.7 %	64.4 %	58.5 %	90.7 %	93.2 %	92.5 %	92.8 %
262	91.1 %	56.2 %	47.6 %	93.0 %	89.8 %	94.3 %	91.3 %

probabilities **A** and, in the next iteration, the mean and deviation values are updated.

In the next step the algorithm checks each mixture component of the emission distributions, if it is preferable to split it in two. The criterion is denoted by

$$\theta_{jm} > \frac{\sum_{j=1}^{N}\sum_{m=1}^{M(j)}\theta_{jm}}{\sum_{j=1}^{N} M(j)},$$

$$\text{with } \theta_{jm} = -\sum_{u=1}^{U}\sum_{t=1}^{T}\ln\left(b_{jm}\left(\mathbf{x}_t^{(u)}\right)\right)\cdot \chi\left[q_t^{(u)*} = j, b_{jm}\left(\mathbf{x}_t^{(u)}\right) = \max_{l}\left\{b_{jl}\left(\mathbf{x}_t^{(u)}\right)\right\}\right]$$

where M(j) is the number of mixture components of state j and U is the number of training sequences. The negative logarithm of a probability, as used in the above equation, is also called *score*. Thus θ_{jm} stands for the sum of all scores, produced by component m of state s_j for all paths $\mathbf{q}^{(u)*}$. In other words θ_{jm} is a measure for the probabilities of component m and the right side of the inequality is the mean of this measure over all components.

Next the procedure decides, if the HMM produces the observation vectors $\mathbf{O}^{(u)}$ with a sufficient probability. We use the criterion

$$\sum_{u=1}^{U}\ln P^{*(n)}\left(\mathbf{O}^{(u)}|\lambda\right) - \sum_{u=1}^{U}\ln P^{*(n-1)}\left(\mathbf{O}^{(u)}|\lambda\right) < \text{const.}$$

where n denotes the iteration number. The algorithm recurses until the difference between the mean score of all U training sequences of iteration n and the mean score of all U training sequences of the previous iteration n-1 falls below a limit.

Finally the algorithm verifies the initial hypothesis for the number of states. In particular, at the beginning and also at the end of a sign there are similar feature vectors that need not be represented by separate states. Thus, states are deleted if the number of vectors that are assigned to a state are not sufficient or if the condition

$$\min_{n,m}\left\{-\ln b_{in}\left(\mu_{jm}\right) - \ln b_{jm}\left(\mu_{in}\right)\right\} < \text{const}$$

is met. All components n of state s_i are compared with all components m of state s_j. If the smallest score sum of the other component falls below a limit, a state is deleted. In this case the parameters must be updated, otherwise the procedure has come to an end and the HMM can be stored for recognition.

4.3 Recognition

If all HMMs λ_i have been trained, signs can be recognised. Using the extracted observation sequence **O** of the sign to recognise the Viterbi procedure sequentially calculates the probabilities $P^*(\mathbf{O}|\lambda_i)$ for all HMMs. Choosing the highest $P^*(\mathbf{O}|\lambda_i)$ the most probable HMM, i.e. sign, can be identified. As with the training, the path probabilities P* must be scaled in order to avoid exceeding the precision range of the computer. Therefore we use the natural logarithm of P* for calculation.

5 Experiments

The experimentation system consists of a CCD video camera and a Pentium PC with an integrated modular image processing system allowing the calculation of the feature vectors at a processing rate of 13 frames per second. In order to ensure correct segmentation, there are few restrictions for the clothing of the signer and the background must have one colour only [6].

The vocabulary of the database consists of 262 signs representing words from ten word types such as nouns, verbs, adjectives etc. The choice of signs used was aimed at simple stories to be told without avoiding minimal pairs. The tests were carried out by two persons, who learned the signs for the experiments. Each person performed signs firstly for the training database and then repeated them for the testing database.

Tab. 2 shows how the different training sets were built. For example Person 1 performed ten samples for each of the 262 signs, i.e. 2620 samples altogether. Set Training 3 in Tab. 2 contains the training data of both persons yielding a total number of 3930 samples. The test sets displayed in Tab. 3 were built in the same way as the training sets, only that Person 1 performed fewer repetitions per sign.

For the recognition task, two additional sub-sets were built of each training set and each test set with a vocabulary size of 43 and 150 of the 262 signs. Tab. 4 shows the results. If Person 1 trains and tests the system, the recognition rate amounts to 98.8%, choosing from 43 different signs. If the system has to decide between 262 signs it still recognises 91.1% of the signs.

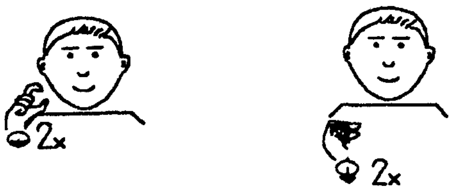

Fig. 3: Minimal pair of the signs EI and EVEN

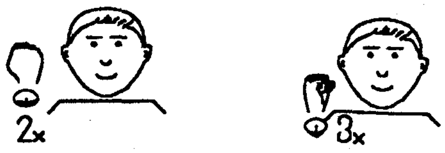

Fig. 4: Minimal pair of the signs STOUT and BONZEN

We obtain similar results if Person 2 trains and tests the system, with rates ranging between 95.8% and 90.7%, depending on the vocabulary size. As expected, the rates fall if one person trains and someone else tests the system. In this case the rates are 56.2% and 47.6%, given a vocabulary of 262 signs. These results can be considerably improved by training the system with both persons. Now the average recognition rate for both signers amounts to 91.3% for 262 different signs.

Analysing the results, it can be stated that the system meets the requirement of signer-dependent recognition. Even most of the minimal pairs can be discriminated such as the signs EI and EVEN, that only differ in the distance of the thumb and the forefinger or such as STOUT and BONZEN, that only differ in the orientation of the dominant hand. Heading in the direction of signer-independent recognition the results prove that the additional training with another person does not significantly decrease the rates of the first person compared to the signer-dependent case, but does drastically increase the rates for the second person. This also proves the feasibility of mixture densities allowing several representations within the same state. Due to the limited training data collected from only two persons, limiting the number of mixtures M=2 is sufficient.

However, the results are encouraging and would suggest that improvements could be obtained. Sometimes errors occur if parts of the two signs are similar. In this case the path stays too long within the states of the similar part and changes to the end state at the very last time steps. Introducing duration distributions for the state could help overcoming these errors. Another problem is repetitions. E.g. the signs KOEK and KOEKJE only differ in the number of repetitions of relatively small motion patterns. Velocity features of the COG of both hands could provide a solution for this case. Further work should also improve the modelling of the non-dominant hand. Features yielding a more detailed description of its handshape must be added. Finally, errors are caused by the reduction of the 3D space on the 2D plane. Either a second camera or special arm models could help reconstructing the third dimension.

6 Summary

We present a video-based system for the recognition of isolated signs with little intrusion on the signer. The system is equipped with a single camera in order to minimise the hardware requirements. We described how the HMM theory adapts to sign language recognition and presented details for the training and recognition procedures. With a feature vector of relatively simple features the results prove that even 262 different signs from two signers can be discriminated with a high probability.

As the results are very promising, the next steps will be a further expansion of the vocabulary. A challenging future task will be the transition to connected sign recognition, allowing more comfortable human-machine interaction.

References

[1] M. Assan, "Videobasierte Gebärdenspracherkennung mit Hidden-Markov-Modellen", *Diplomarbeit*, Lehrstuhl für Technische Informatik, Aachen, 1997.

[2] P. Boyes Braem, *Einführung in die Gebärdensprache und ihre Erforschung*, Signum Verlag, Hamburg, 1995.

[3] D. Brentari, "Sign Language Phonology: ASL". *Handbook of Phonological Theory*, Basil Blackwell, New York, 1994, pp. 615-637.

[4] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick and A. Pentland, "Invariant features for 3-D gesture recogniton", *Proc. of the 2. Int'l. Conf. on Autom. Face & Gesture Rec.*, Killington, 1996, pp. 617-621.

[5] Y. Cui and J.J. Wenig, "View-based Hand Segmentation and Hand-Sequence Recognition with Complex Backgrounds", *Proc. of ICPR*, 1996, pp. 617-621.

[6] K. Grobel and H. Hienz, "Video-based recognition of fingerspelling in real-time", *Proc. des Workshops Bildverarbeitung für die Medizin*, Aachen, 1996, pp.197-202.

[7] K. Grobel und H. Hienz, "Videobasierte Erkennung von Körperregionen zur Bestimmung der Ausführungsstelle einer Gebärde", *Proc des 9. Aachener Kolloquiums Signaltheorie*, März 18-20, 1997.

[8] H. Liang and M. Ouhyoung, "A sign language recognition system using Hidden Markov Model and Context Sensitive Search", *ACM VRST*, 1996.

[9] R. Rabiner, "A Tutorial on Hidden Markov Models and selected applications in speech recognition", *Proc. of the IEEE*, vol. 77, no. 2, 1989, pp. 257-285.

[10] J. Schlenzig, E. Hunter and R. Jain, "Video Based Hand Gesture Interpretation Using Recursive Estimation", *Proc of ACSSC*, Pacific Grove, Nov. 1995, pp.1267-1271.

[11] E.G. Schukakat-Talamanzini, *Automatische Spracherkennung*, Vieweg Verlag, Braunschweig, 1995, pp. 121-164.

[12] T.E. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models", *Technical Report no. 375*, MIT Cambridge, Media Laboratory, 1995.

[13] J. Yamato, J. Ohya and K. Ishii, "Recognizing Human Action in Time-Sequential Images Using Hidden Markov Model", *Proc. of the IEEE Conf. on Comp. Vision and Pattern Recognition*, 1992, pp. 379-385.