

Automatic Recognition of Dutch Sign Language

Keywords: sign language, automatic recognition, classification, pattern recognition, markov model

G.A. ten Holt*‡, P. Hendriks†, T.C. Andringa‡, E.A. Hendriks*, M.J.T. Reinders*

* ICT Group, dept. of Mediamatics, Delft University of Technology

† Center for Language and Cognition Groningen, University of Groningen

‡ Auditory Cognition Group, dept. of Artificial Intelligence, University of Groningen.

Abstract

This paper describes the recognition of isolated Dutch sign language signs. Data is gathered with one digital camera. Head and hands are tracked, and the values are categorised to produce a linguistically inspired feature vector. With this feature vector, classification is performed. We build left-to-right Markov chain models without skips or loops to represent single signs. As an alternative, we use a classification method that compares vectors present in a sign, without regard for order. Due to the strictness of the left-to-right requirement, classification with the Markov models often fails. Classification with the alternative method yields reasonable results, supporting the notion that the linguistic feature vector is suitable for classification. More knowledge on morphology of signs would help to make more appropriate models for sign recognition.

1 Introduction

Sign languages are natural languages that develop spontaneously wherever groups of deaf people come into contact. They operate in the visual-spatial domain instead of in the aural-oral domain of spoken languages. This makes them of scientific interest: do our theories on language hold for sign languages as well?

Research into automatic recognition of sign languages serves several purposes: it can be used to develop communication aids for sign language users, but also as a first step towards more general gesture recognition techniques that can be used in human-computer interaction.

Automatic sign recognition broadly consists of four phases: data acquisition, data encoding, classification, and translation [2]. Acquisition is concerned with capturing the sign data. This can be done by camera(s) or with the aid of other sensors, such as magnetic trackers. Encoding is about finding the information important for sign recognition and storing it in a certain format. Identifying the correct sign based on this information is called classification. Finally, a translation step is necessary when continuous signing is recognised: because sign languages are separate languages, a word-by-word translation to another (spoken) language is not enough to reflect all information.

In this paper, automatic recognition of Dutch sign language (NGT) is investigated. We have reviewed several existing methods of sign recognition. Based on their results as well as their usability, we have chosen to implement the Linguistic Feature Vector method [1]. Our main interest goes out to the classification part of the recognition. We wish to

investigate how well we can model Dutch signs with the information contained in the linguistic feature vector. Translation was not necessary, because we worked with isolated signs. We used digital video to capture our data and used coloured gloves to facilitate the detection of the hands. We only investigated recognition of isolated Dutch signs from a single user.

The next section gives a short overview of the current techniques in automatic sign recognition (ASLR) and their merits. In section 3, our method is described in detail. The experimental setup is described in section 4, and the results are given in section 5. Section 6 presents a short discussion.

2 Previous Work

Automatic sign recognition has been investigated since around 1995 [3]. Researchers tried a variety of techniques, such as fuzzy logic [4], neural networks [3, 5], and Hidden Markov Models (HMMs) [6-9], mostly on small vocabularies. For a detailed overview of techniques and results, we refer to [10]. Vogler and Metaxas [11] achieved a 96% recognition rate on a 22-word vocabulary using parallel HMMs. Chen et al. [12] reported a 92% success rate with whole-word HMMs for a 5113-word vocabulary. Both used instrumented gloves and magnetic trackers to capture their data. Zieren and Kraiss [13] and Bowden et al. [1] use cameras to record signs. They achieve 98% recognition rates for a 152- and a 43-word vocabulary respectively. Zieren and Kraiss use whole-word HMMs, whereas Bowden et al. use Markov chain models to recognise signs. Our preference went out to a vision-based method. First, with vision-based methods signers do not

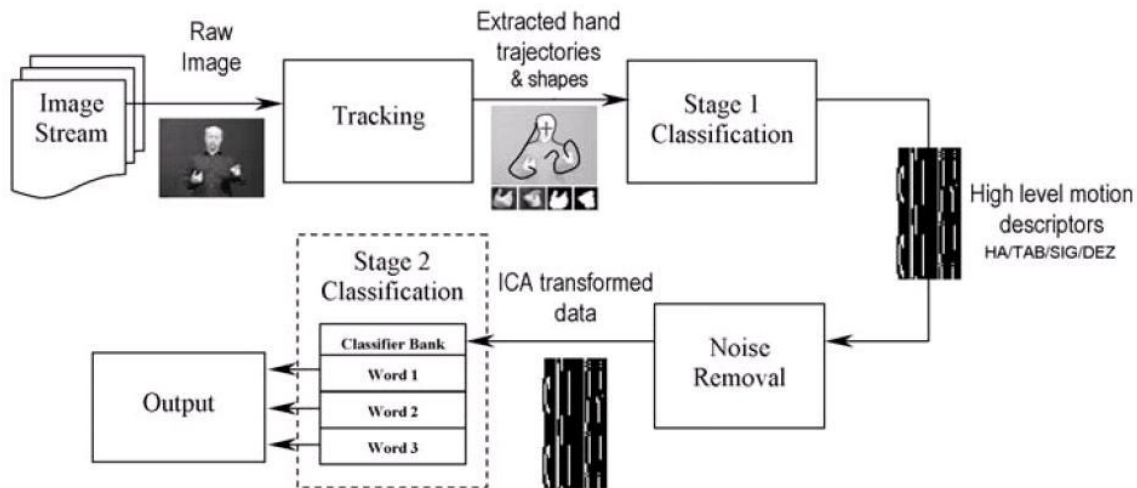


Figure 1: Overview of the Linguistic Feature Vector Method. Taken from [1]

have to wear hardware (sensors and trackers) on their bodies, so they are less encumbered. Secondly, vision-based methods have access to the face, as opposed to sensor-based methods. Facial information is important in sign languages, therefore, vision-based methods have more future perspectives. The vision-based method of Bowden et al. achieved good results with Markov chain models. Successful models could be built from as little as a single training example, whereas HMMs need many examples per model. For these reasons, we chose to base our recognition of Dutch sign language on this method.

3 Method

Our method operates as follows: First, data is captured on digital video. Then classification proceeds in two phases. In the first phase, the hands are tracked and a generalisation over position and movement is executed. This is achieved by dividing individual values for position and motion into categories, and using these categories as features. The categories reflect distinctions that are significant in sign language. This first phase is called stage I classification.

With the resulting feature vectors, a model bank of signs is built from training data. An unknown observation can then be identified by finding the model that is most likely to produce this observation. This is called stage II classification. The models used are Markov chains. They can be built successfully from as little as a single training example. An overview of the recognition process is shown in Figure 1. We will now discuss the phases of the method in some detail.

3.1 Tracking

Our main interest lies in the classification part of automatic recognition. Therefore, tracking is not the

focus of this paper, but for completeness, we will briefly describe it here.

Visual tracking is performed as follows: probabilistic labelling of the skin is used to roughly locate the signer's face. A contour model of head and shoulders is then fitted to the signer. This outline provides a body-centred co-ordinate system. Key body locations (face, stomach, left shoulder etc.) are determined relative to this outline. The hands are found using skin labelling (we used coloured gloves to make it easier) and their position and velocity are expressed relative to the body-centred co-ordinate system.

3.2 Stage I Classification

In this stage, raw image data is transformed into a sequence of feature vectors, one vector for each frame. Linguistic evidence suggests that recognition of a sign is primarily based on the information transmitted by the dominant hand [1]. For this reason, the non-dominant hand is ignored for now and only information from the dominant hand (in our case the right hand) is extracted. The extracted features are:

Hand Arrangement (HA): This value can be computed directly from the x- and y-positions of the centres of the hands and their approximate area (in pixels).

Position (TAB): This value is determined according to the proximity of the hands to the key body parts, estimated in the tracking-stage. As a distance metric, Mahalanobis distance is used.

Movement (SIG): To keep noisy motions from interfering with the detection of real movement, the size of the hand is used as a threshold for movement. This entails that small movements such as finger-wiggling and rotations of the wrist cannot be

detected. All possible values for the SIG-parameter are large movements.

The possible feature values are shown in Table 1. For each frame, the HA, TAB and SIG-features are extracted and stored in one binary vector. In this manner, stage I classification translates a movie (a series of frames) into a series of 33D binary vectors containing mostly zeros. Figure 2 gives an example of such series of binary vectors.

Hand Arrangement (HA)	Position (TAB)	Motion (SIG)
– right hand high	– neutral space	– hand makes no movement
– left hand high	– face	– hand moves up
– hands side by side	– left side of the face	– hand moves down
– hands are in contact	– right side of the face	– hand moves left
– hands are crossed	– chin	– hand moves right
	– right shoulder	– hands move apart
	– left shoulder	– hands move together
	– chest	– hands move in unison
	– stomach	
	– right hip	
	– left hip	
	– right elbow	
	– left elbow	

Table 1: Overview of possible feature values. These three features concatenated make up the linguistic feature vector.

Note that within a feature more than one value can be active for the same frame. It is not clear whether this is a way of encoding information (for instance representing diagonal movement by making both ‘forward movement’ and ‘sideways movement’ active), or whether it is a matter of values

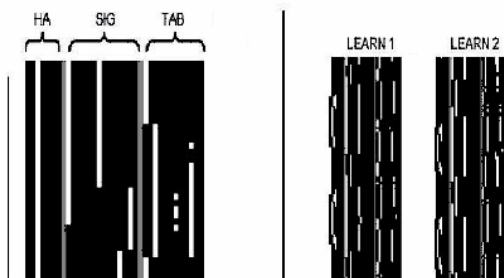


Figure 2: Example of signs represented with binary feature vectors. The vectors are depicted horizontally. Black is 0, white is 1. The gray lines mark the boundaries between parameters. Taken from [1].

competing and both being active enough to come above a threshold (for instance when position is on

the edge of ‘stomach’ and ‘right hip’ and both these values are therefore considered active).

The benefit of first dividing tracker values in categories, instead of using them immediately for classification, is that in categorising, unimportant variation is discarded. By choosing the category boundaries correctly according to sign linguistics, it is ensured that variation within a category is not important for the meaning of a sign. It is therefore enough to retain only the category labels for the various features.

3.3 Stage II Classification

After the characteristic information has been extracted from the image data in stage I classification, stage II classification can begin. First, there is a training phase in which models are built from the training signs’ feature vectors. For each sign in the vocabulary, one model is built. After this training phase, an unknown sign can be classified by finding the model that represents it best. Before the models are built, independent component analysis (ICA) is performed to enable comparison between the feature vectors.

3.3.1 Independent Component Analysis

Before we start modelling with the extracted linguistic feature vectors, we want to cluster them, to be less sensitive to noise and small tracking errors which can cause vectors that should be identical to differ. But our feature vectors are not easy to cluster, because you cannot use a simple distance metric to judge the similarity of two binary vectors.

Therefore, we use ICA to transform the binary vectors. ICA is used to find the independent components of a mixture. In our method, the feature vectors are regarded as a mixed signal, and ICA is used to find the independent components and the unmixing matrix. By transforming the binary vectors with the unmixing matrix, they are transformed to a space where dependent vectors come to lie close together, and unrelated ones further apart. After this transformation, the vectors can be clustered using a Euclidian distance metric. Each cluster is appointed a symbol. These symbols are used as the state observations in the models. Figure 3 shows the transformation from binary vector to symbol.

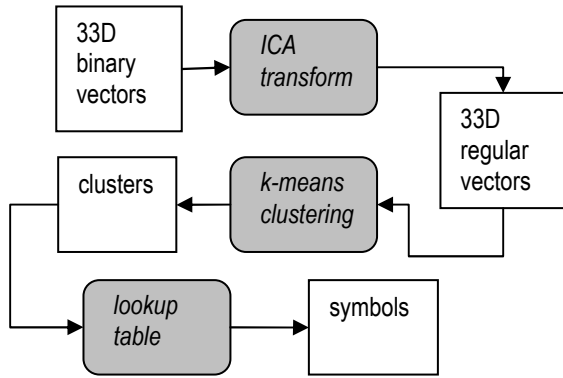


Figure 3: Overview of the process from binary feature vector to symbol

3.3.2 Modelling

The models we use are first order left-to-right Markov chains. They consist of a number of states with one possible observation in each state, and transitions between these states with certain probabilities. Models are built for every sign in the vocabulary using only a single training example per sign. First, a sign is transformed to a series of symbols in accordance with Figure 3. Then a state is created for each new symbol in the series. The only transitions allowed are the auto-transition and the transition to the next state. Transition probabilities are calculated by expressing the proportion between both types of transition as fractions of 1.

For example: take the sign represented by the symbol series AAABCA. For this sign, a model with four states is created. The state observations are A, B, C, and again A. As for the transitions: for the first state, the auto-transition has a probability of 0.67, the transition to ‘B’ a probability of 0.33. The rest all have a transition-to-next-state probability of 1. Figure 4 shows a symbol series and the model that is created from it.

Our model design allows no skips, loops or backward transitions. That means a model is required to go through all its states in the proper order. This requirement is taken from speech recognition and adopted to avoid transitions that cause the model to respond to symbol series it was not intended for.

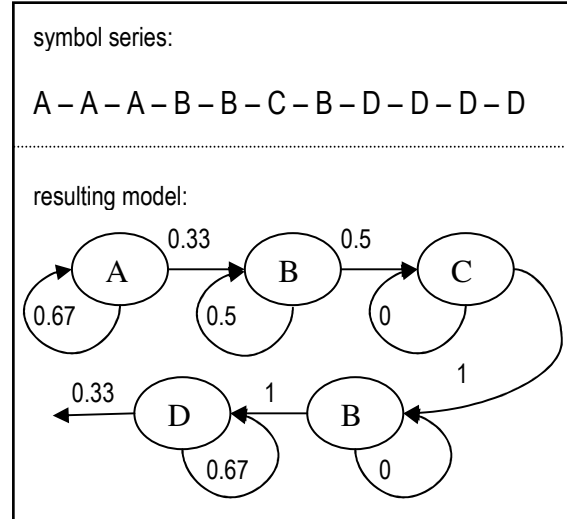


Figure 4: Building a model from a series of symbols. The frequency of a transition in the series is used to calculate the transition probability.

After the model bank is built, unknown signs can be classified. This is done by first extracting the feature vectors from the sign movie of the unknown sign, and then transforming the resulting binary vectors with the ICA-matrix that was found in training. The transformed vectors are translated into symbols by finding the cluster centroid (from the training set clusters) a new vector resembles most, and using the symbol associated with that cluster. Simple Euclidian distance is used as a distance metric.

After the new sign is thus represented as a series of symbols, it can be calculated which model w has the greatest probability of producing this symbol series s . We know that

$$P(s | w_i) = \prod_{t=1}^m P_{w_i}(s_t | s_{t-1})$$

Where

- w = model
- s = symbol series
- m = length of s
- $P_w(s_t | s_{t-1})$ = transition probability from the current state to the next state needed

Using Bayes’ rule, we can state that

$$\arg \max_i P(w_i | s) = \arg \max_i P(s | w_i)P(w_i)$$

Assuming equal prior probabilities for all signs in the vocabulary (that is, equal $P(w_i)$), finding the

model w with the greatest probability of producing s becomes a matter of finding $\arg \max_i P(s | w_i)$.

3.3.3 Alternative Classification Method

Because the strictness of the left-to-right models might cause problems, an alternative method of classification was also investigated. Here, a simple comparison was made between the symbols (representing feature vectors) in the unknown sign, and those in the various models. If s_{model} is the set of unique symbols in the model and s_{obs} is the set of unique symbols in the observation sequence, then the likeness score was calculated as follows:

$$score_{model} = \frac{size(s_{model} \cap s_{obs}) - |size(s_{model}) - size(s_{obs})|}{size(s_{model})}$$

The score gets better when model and unknown sign have more symbols in common and worse when the difference in set size is greater. Note that no demands are made as to the sequence of the feature vectors. The recognition result was the sign represented by the model that got the highest score.

4 Experimental Setup

To test this method on Dutch sign language, a movie containing a number of NGT signs was recorded. The movie was shot against a matt black background. The signer wore a brightly coloured reddish-pink glove on his dominant (right) hand, and a white one on his other hand. He wore dark clothing and stood straight in front of the camera, visible from head to hips. The signer was a male native NGT signer (born deaf with deaf parents) aged 25. The movie was recorded with a Panasonic NV DX100 digital video camera at 25 fps. This sampling rate is high enough to ensure that no important parts of the sign can get lost between frames.

The experiment used 31 isolated NGT signs. They were chosen randomly. Each sign was performed slowly 6 times, so a total of 186 signs was recorded. Between the signs, the hands returned to a neutral rest position (hands clasped in front). The vocabulary words are included in Appendix I. The training set consisted of 2 examples per sign, a total of 62 signs. The other 4 examples formed the test set, a total of 124 signs.

We made two models per sign (one for each sign in the test set). Recognition was deemed correct when

at least one of these two models had the highest score.

We also tested recognition with the alternative classification method described in section 3.3.3. Again, recognition was called correct if at least one of the two models for the correct sign had the highest score.

5 Results

Bowden et al. [1] reported a success rate of 98% for a 43-word vocabulary. However, this was after removal of six signs that were indistinguishable to the system because they only differed in facial features (which were not extracted). For the original 49-word vocabulary, recognition was 84%. Furthermore, Bowden et al. used a trial-and-error-method to remove certain ICA-components. (The reasoning being that since noise is independent of the signal, a number of the independent components found in ICA will represent noise. Those components whose removal caused the greatest improvement in recognition results were considered noise and removed.) Without this noise removal, their recognition rate was 73%.

Because of time limitations, we did not implement the noise removal step. We also did not remove any signs from our vocabulary.

Our test results were as follows: when testing with the strict models and thus demanding precisely the same sequence of feature vectors, we achieved a recognition rate of 35% for the 124-sign test set. Recognition was deemed correct when at least one of the top scoring models was the correct sign. Since there were cases in which different models (that is, models representing different signs) had the same top score, the recognition rate would have been lower if these cases had been rejected. However, those top scoring models were often totally identical – for example in the case of models representing signs which only differed in handshape (we did not have a handshape feature). Therefore, it can be debated whether it would have been fair to the system to reject those cases.

When the alternative classification method of section 3.3.3 was used, the recognition rate rose to 65%. However, once again recognition was counted as successful when at least one of the models with the highest score was the correct one. Ties occurred frequently in this setup, due to the simple nature of the comparison method, so the score would have been lower if only unique identifications had been counted as correct.

6 Discussion

The results of the classification are low compared to the results Bowden et al. obtained with the same method: 35% versus 73%. There are a number of factors that could explain this difference, such as technical differences in implementation. The most

likely explanation, however, is the strictness of the models used. We built our models according to the strict rule that every state must be visited, and must be visited in the right order. During training and testing, though, it became obvious that a ‘wrong’ symbol often sneaks into an otherwise identical symbol series. This could be due to variation in signing, or even to noise.

If it were an option to skip such a state, the model would work. But because of the requirement of strict equality, skipping is not an option and the model fails. Since Bowden et al. do allow some skipping of states, this could explain why their models are more often successful than mine. An example of ‘wrong’ vectors messing up a model can be seen when comparing the symbol sequences of two examples of IK (“I”) in Figure 5 (the numbers in row and column headers are the symbols). In the first model, a single aberrant vector (represented by symbol ‘C’) was apparently present, causing the two examples to have different symbol sequences.

The sign IK is made by pointing at the chest. One could imagine that in the first instance of the sign, the signer happened to raise his hand a bit higher at one moment during the sign, and one vector consequently received the aberrant position value ‘chin’ instead of ‘chest’. This resulted in a different vector, which received the symbol ‘C’, whereas all the other vectors received the symbol ‘B’. And this causes the symbol sequences to differ. If skipping such a state were an option, models would be more robust.

	A	B	C	B
A	0.67	0.33		
B		0.67	0.33	
C			0	1
B				1

(1)

	A	B
A	0.75	0.25
B		1

(2)

Figure 5: Two models for the sign IK (“I”). The letters in row and column header are the state symbols, the table cells are the transition probabilities from state <row> to state <column>. (So the transition probability from ‘A’ to ‘B’ in model 1 is 0.33.)

The two models are different only because a single aberrant feature vector (represented by the symbol ‘C’) is present in the first model.

The models built by Bowden et al. not only allow skips, they allow loops as well. This is another example of model flexibility which is very useful

for signs. Many signs are periodic. This means that they consist of a certain movement made a variable number of times. Since our models are linear, a model ‘A-B-A-B’ cannot produce the series ‘A-B-A-B-A-B’, though both these series are the same sign: a periodic sign making first two, then three cycles.

	A	B	A	B	A	B	A
A	0.88	0.1					
B		0	1				
A			0.83	0.17			
B				0	1		
A					0.88	0.12	
B						0	1
A							1

(1)

	A	B	A	B
A	0.95	0.05		
B		0.75	0.25	
A			0.90	0.10
B				0

(2)

Figure 6: Two models for the sign VANDAAG (“today”). Each model can only produce the observation sequence for a sign with exactly that number of repetitions. So model 2 can only produce A-B-A-B, not A-B-A-B-A-B (one repetition extra). This shows why periodic signs need more sophisticated models.

Allowing a loop helps a model to produce these different variants of a periodic sign. For example, the models in Figure 6 clearly represent the same sign. But if it does not contain a loop, one model cannot produce different variants of the sign. However, it is not trivial to decide which skips and loops should be allowed in a model, and when they should be added. If in one training example state ‘B’ is skipped and in another state ‘D’, and both transitions are added, the model can then also produce a symbol sequence in which both these states are absent. And this may not be allowed in the sign. Similar problems exist for building loops into models. Unfortunately, the rules governing the form and allowed variations of signs (especially periodic signs) are often still unknown for Dutch signs. This makes it difficult to determine what can and cannot be allowed in a sign model.

When the requirement of strict sequence equality was abandoned and a crude method of comparing the symbol sets of the unknown sign and the models was used, a recognition rate of 65% was

achieved. This compares better to the 73% of Bowden et al. It strengthens the hypothesis that the low original success rate of 35% is mainly due to the strictness of the rule about sequence equality, and that the extracted features are suitable for classifying signs (65% is a lot better than chance). And given the crudeness of the comparison method used and the fact that handshape information could not be used, the recognition rate could probably be higher. One possible way of improving recognition would be to use language models which can predict the probability of one sign following another. In this experiment, however, we performed isolated sign recognition only, so this method was not relevant.

The difference between British sign language (BSL) and Dutch sign language probably does not play a part in the difference in success rate. In BSL as well as in NGT, the parts of a sign that determine its meaning are: handshape, hand position, hand orientation, hand movement and the non-manual component [14, 15]. Extracting information on hand position, hand movement and arrangement of the two hands and using this to classify a sign should therefore work no better or worse for NGT than it does for BSL. However, some reservation on this point may be wise.

In conclusion, we can state that the most probable cause of the low recognition rate is the strictness of the models used. The requirement that every state in the model must be visited in the correct order for a model to be successful is obviously problematic. This requirement is taken from automatic speech recognition, and for this field it is a reasonable restriction. But spoken languages do not possess periodic words. Because periodicity is possible in sign languages, loops in models may be a necessity, and the same is true for certain skips and backward transitions. Model design depends greatly on what variation is allowed in a sign. More theoretic information on the nature and composition of signs will help to draw up better rules for building sign models.

References

1. Bowden, R., et al., *A linguistic Feature Vector for the visual Interpretation of Sign Language*, in *Proceedings of the 8th European Conference on Computer Vision ECCV'04*, T. Pajdla and J. Matas, Editors. 2004, Springer-Verlag, Heidelberg: Prague, Czech Republic. p. 391-401.
2. Braffort, A., *Research on Computer Science and Sign Language: Ethical Aspects*, in *Gesture and Sign Language in Human-Computer Interaction: Proceedings of the International Gesture Workshop GW2001*, I. Wachsmuth and T. Sowa, Editors. 2002, Springer-Verlag, Heidelberg: London, UK. p. 1-8.
3. Waldron, M.B. and S. Kim, *Isolated ASL sign recognition system for deaf persons*. *IEEE Transactions on Rehabilitation Engineering*, 1995. **3**(3): p. 261-271.
4. Holden, E.-J., R. Owens, and G. Roy, *Adaptive fuzzy Expert System for Sign Recognition*, in *Proceedings of the International Conference on Signal and Image Processing SIP'2000*. 1999: Las Vegas, USA. p. 141-146.
5. Vamplew, P. and A. Adams, *Recognition of Sign Language Gestures using neural Networks*. *Australian Journal of Intelligent Information Processing Systems*, 1998. **5**(2): p. 94-102.
6. Vogler, C. and D. Metaxas, *Adapting Hidden Markov Models for ASL Recognition by using three-dimensional Computer Vision Methods*, in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics SMC97*. 1997, IEEE Computer Society: Orlando, Florida. p. 156-161.
7. Starner, T., J. Weaver, and A. Pentland, *Real-time American Sign Language Recognition using desk and wearable Computer based Video*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998. **20**(12): p. 1271-1375.
8. Liang, R.-H. and M. Ouhyoung, *A Real-Time Continuous Gesture Recognition System for Sign Language*, in *Proceedings of the Third International Conference on Automatic Face and Gesture Recognition*. 1998: Nara, Japan. p. 558-565.
9. Grobel, K. and M. Assam, *Isolated Sign Language Recognition Using Hidden Markov Models*, in *the IEEE International Conference on Systems, man and Cybernetics*. 1997: Orlando, FL. p. 162-167.
10. Ten Holt, G., P. Hendriks, and T. Andringa, *Why don't you see what I mean? Prospects and Limitations of current automatic Sign Recognition Research*. *Sign Language Studies*, accepted for publication.
11. Vogler, C. and D. Metaxas, *Handshapes and Movements: Multiple-Channel American Sign Language Recognition*. *Lecture Notes in Computer Science*. 2004. 247-258.
12. Chen, Y., et al., *CSLDS: Chinese Sign Language Dialog System*, in *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and*

- Gestures AMFG'03*. 2003, IEEE Computer Society: Nice, France. p. 236-238.
13. Zieren, J. and K.-F. Kraiss, *Non-Intrusive Sign Language Recognition for Human-Computer Interaction*, in *9th IFAC/IFIP/IFORS/IEA Symposium Analysis, Design, and Evaluation of Human-Machine Systems*. 2004: Atlanta, GA.
 14. Schermer, G.M., et al., *De Nederlandse Gebarentaal*. 1991, Twello: Van Tricht.
 15. Deuchar, M., *British Sign Language*. 1984, London, UK: Routledge & Kegan.
 16. Zeshan, U., *Interrogative Constructions in signed Languages: crosslinguistic Perspectives*. *Language*, 2004. **80**(1): p. 7-39.

Appendix I: Dutch sign language vocabulary

Vocabulary word:	English translation:
ik	I
werken	to work
kantoor	office
gaan	to go
vandaag	today
niet	not
ziek	ill
maar	but
morgen	tomorrow
vrijdag	Friday
als	if
beter	better/well
weekend	weekend
ons	our
gezellig	convivial
collega	colleague
wij	we
kletsen	to chat
veel	much/a lot
ook	also
1	one (1)
hij	he
oud	old
moeder	mother
man	man
hele	entire
dag	day
andere	other
praten	to talk
mogen	to like
daarom	that's why

Note: the word 'he' appears in the vocabulary. This is actually not a fixed sign, but one that differs with context, as is the case for all third-person pronouns in NGT [14], and probably in all sign languages [16]. The word was included in one possible form (pointing to the left) as if it were a fixed sign, so that it could be used in a story, which was recorded after the vocabulary signs. It was necessary to treat 'he' as a fixed sign because the method implemented here cannot handle signs that differ with context, and it would have been difficult to tell a story without the use of a third person pronoun. In the end, the sign language story was not used in our experiment.