Diplomarbeit im Fach Informatik
Rheinisch-Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik 6
Prof. Dr.-Ing. H. Ney

Appearance-Based Features for Automatic Continuous Sign Language Recognition

vorgelegt von:
David Rybach
Matrikelnummer 229337

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. Dr. J. Borchers

Betreuer:
Dipl.-Inform. T. Deselaers
Dipl.-Inform. P. Dreuw

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, im Juni 2006

David Rybach

Abstract

This diploma thesis investigates appearance-based features for the person-independent vision-based recognition of continuous sign language. A large variety of methods which have been successfully used for automatic speech recognition is applied to this task. Appearance-based approaches do not rely on a segmentation of the images or on predefined models of the image content and use the image itself as the feature. A novel tracking algorithm is introduced and applied to hand and head tracking. The tracked body parts are used in order to calculate additional features to improve recognition performance. The presented automatic sign language recognition system is evaluated on a set of sentences in American Sign Language.

Acknowledgment

I would like to express my gratitude to all people who supported me during the progress of this work. Especially I would like to thank:

Prof. Dr.-Ing. Hermann Ney for the interesting possibilities at the Chair of Computer Science 6 of the RWTH Aachen University and for many helpful discussions,

Prof. Dr. Jan Borchers who kindly accepted to co-supervise this work,

Thomas Deselaers and Philippe Dreuw for the supervision of this work and for their countless ideas and suggestions,

Morteza Zahedi for the collection and the transcription of the RWTH-Boston-201 database,

Daniel Stein for the introduction to sign language and for the annotations of the RWTH-Boston-104 database,

Stefan Hahn and Arne Theres for many helpful tips, comments and discussions about speech recognition,

Daniel Schneider for proof reading the manuscript and a lot of helpful suggestions,

Julia Klubert who helped me annotating the RWTH-Boston-Hands database,

and finally my parents for supporting me in every way.

Contents

Chapter 1

Introduction

This diploma thesis presents a system for the automatic recognition of continuous sign language. Different features describing visual and temporal aspects of signs and the usage of these features for sign language recognition are investigated.

Sign language is used by deaf people and people being hard of hearing for communication. It is a structured form of visual communication, comparable to spoken languages in complexity and expressiveness. Approximately 80,000 deaf people live in Germany, most of them speaking German Sign Language (Deutsche Gebärdensprache, DGS). German Sign Language has been recognized legally in 2002 and can be used for the communication with public authorities (Gesetz zur Gleichstellung behinderter Menschen § 6). However, in most cases the direct communication between deaf and hearing people is difficult, because either a sign language interpreter has to mediate or written communication has to be used.

Recent progress in image processing [Huang & Sebe$^+$ 06], speech recognition [Gauvain & Lamel$^+$ 05], and machine translation [Ney 05a] gives a prospect of using computers to bridge this communication barrier by automatically translating sign language in spoken language and vice versa. Automatic sign language recognition plays an important role in this – currently still academic – translation system. Systems for the automatic recognition of sign language aim at recognizing single signs or sequences of signs in continuous signing. The translation of the recognized signs to a spoken language can be done by an additional translation system [Stein & Bungeroth$^+$ 06] or by coupling recognition and translation [Ney 99]. In the statistical approach to automatic sign language recognition, the structure of signs and language is modeled by probabilistic knowledge sources, which enables the system to learn automatically without human intervention.

Research in the area of sign language recognition is challenging, because it requires methods known from gesture recognition, speech recognition, and image processing. Automatic sign language recognition can be applied, for example, to the following tasks:

Translation. Translation systems can aid the communication between deaf and hearing people.

Sign language learning. An application to teach sign language should control the progress of the user by examining the performed signs.

Transcription of videos. Transcriptions of sign language videos are needed for different kinds of further processing, for example sign language translation. Currently, transcribing sign language videos, i.e. writing down the performed signs, is usually done manually, which is a rather simple but nevertheless very costly process. The transcription proposed by the automatic recognition system could be modified by a human operator afterwards.

Dialog systems. Dialog systems and information systems are usually operated by speech and are thus unusable for deaf people. Interfaces for barrier free dialog systems should allow sign language input.

In current research systems, different types of input data are used: Some systems use data gloves or motion capturing systems to measure the position and movements of body parts directly. In contrast, vision-based systems use images captured by a video camera, which is less intrusive and does not require complex and expensive hardware. Features of the signs have to be extracted from the captured sequence of images.

The focus of this work is set on appearance-based features for vision-based recognition of continuous sign language. Appearance-based features are obtained directly from the input images without a feature extraction of segmented image parts. Additionally, features describing special aspects of signs are investigated in this work. For the extraction of these features, new methods for hand and head tracking in sign language videos are developed. Methods that have been investigated successfully in speech recognition are applied to sign language recognition.

The main contributions of this work are quantifiable analyses of several features, different tracking methods, and a large number of model parameters. A sign language recognition system has been implemented based on the RWTH speech recognition system developed by the Human Language Technology and Pattern Recognition group of the RWTH Aachen University. Furthermore, a set of tools for the visualization and analysis of recognition and tracking results has been developed.

This document is organized as follows: Chapter 2 gives an overview on sign language, its grammar, and commonly used notation systems. Chapter 3 describes the underlying theory of sign language recognition and gives a survey on available systems for the recognition of isolated and continuous sign language. In Chapter 4, the tracking framework and the used methods for hand and head tracking are presented. Chapter 5 describes appearance-based features, features for specific aspects of sign language, and their combinations. The databases used for the evaluation of the features and the tracking methods are presented in Chapter 6. Results of the experiments with different features and various tracking methods are shown and discussed in Chapter 7. Finally, Chapter 8 gives a conclusion of this work and perspectives for further research.

Chapter 2

Sign Language

Deaf persons use a sign language for communication with each other. Sign languages are self-contained languages with their own structure and grammar. They have developed in a natural way in deaf communities and independent from spoken languages. Therefore sign language is not universal. Instead, a different type of sign language has evolved in every deaf community. Sign languages can be grouped into national languages, e.g. American Sign Language (ASL) and German Sign Language (Deutsche Gebärdensprache, DGS), which can have several dialects again.

Sign language is frequently confused with pantomime, where the whole body is used to illustrate the meaning of single words or phrases. Some signs in sign language (iconic signs) are illustrative, too. However, pantomime is limited to concrete subjects, whereas sign language is not limited in expressiveness.

Research in sign languages is relatively young. The first serious investigations started about forty years ago [Stokoe 60]. Most of the work has been done in linguistics, but in the last few years research in statistical sign language translation [Bungeroth & Ney 04, Stein 05] and sign language recognition has been done as well. A detailed discussion on past and ongoing research in sign language recognition is given in Section 3.6.

In contrast to spoken languages, which use acoustic signals, sign language uses visual communication. The devices for visual communication can be divided into two groups: manual components and non-manual components.

2.1 Manual Components

In [Stokoe & Casterline$^+$ 65] it is stated that there are three manual components: hand configuration, place of articulation and movement. These components were extended by hand orientation as fourth component [Klima & Bellugi 79, Battison 78].

Hand configuration. The hand shape, i.e. the positions of the fingers, is called hand configuration. Due to the high degree of freedom of the fingers, a lot of hand shapes are possible, but not all of them are used in sign language. Which configurations are used, differs from language to language. However, six basic configurations can be found in most languages [Battison 78] (see Figure 2.1). Each

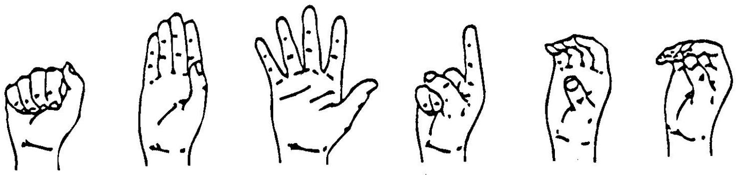


Figure 2.1. Basic hand shapes (from [Braem 95])

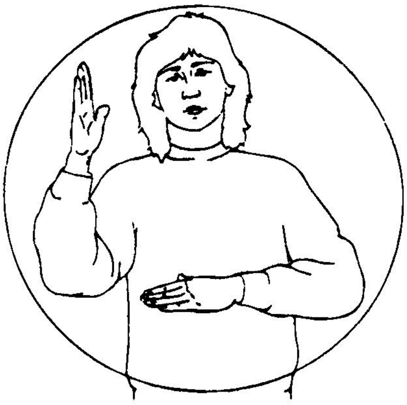


Figure 2.2. The signing space (from [Braem 95])

language defines additional hand shapes. For example, in DGS there are 34 additional configurations.

Place of articulation. The location in signing space, where the sign takes place, is referred to as place of articulation. The *signing space* is the area in front of the signer in which the signs are performed. It includes the face and the upper part of the body (see Figure 2.2). Places of articulation can be grouped into lexical meaningful areas. The area in front of the face is used often, because signing persons look in the eyes of each other not on the hands while they are "listening". The size of a sign can change depending on the "loudness". Whispered signs are smaller than shouted ones.

Hand movement. Hand movement refers to the trajectory of one or both hands in space. Some signs are performed with only one hand, called the *dominant* hand. In signs performed with two hands the non-dominant hand performs a similar movement as the dominant hand or has a supporting role. Deaf people recognize many signs already using only the hand movement [Braem 95].

Hand orientation. The hand orientation is defined by the situation of wrist, elbow, and shoulder. Signs with equal hand configuration can differ in hand orientation. Differences in hand orientation become apparent by different views of the palm.

A pair of signs can be found for each component, such that both signs differ only in this component. The combination of manual components is subject to (national) rules. Not all combinations are allowed, similar to spoken languages, where not all possible phoneme combinations are used.

2.2 Non-Manual Components

Non-manual components include facial expression, eye gaze, head tilt and body posture. They are used for flection and other grammatical purposes. For example, facial expression and head position can be used to indicate questions, negations, and subordinate clauses [Sandler 99]. One of the first studies on non-manual components showed that it is possible to obtain some information about the topic of a conversation if the hands are not visible [Baker & Padden 78].

2.3 Grammar

Sign language grammar differs to spoken language grammar to a large extent. For example, in DGS the verb is always placed at the end of a sentence. Furthermore, there are no articles and copula.

In spoken language, speech has a linear structure: A sentence is a sequence of words, and a word consists of a sequence of sounds. Also in sign language, signs are produced one after the other to formulate a sentence. However, in sign language sub-lexical units can occur simultaneously. Hand movement, hand shape, place of articulation, and non-manual signals can be produced simultaneously and independent from each other. One reason for this simultaneity is that the eye can capture more simultaneously occurring events than the ear [Braem 95].

The use of the spatial dimension, in addition to the temporal one, gives supplementary possibilities to convey meaning. Spatial information is used to flex and to derive words. Flexed verbs consist of the same basic gesture but differ in movement speed, direction, and expansion. Figure 2.3 shows examples for inflection.

Another unique feature of sign language is *indexing*. Objects can be positioned in signing space and then used again by pointing to them [Wrobel 01]. For example, unknown words, like names or foreign words, are spelled (using a finger alphabet) and placed in signing space to avoid spelling them again. Absent persons can be described and indexed similarly. Temporal information is also transfered with spatial principles.

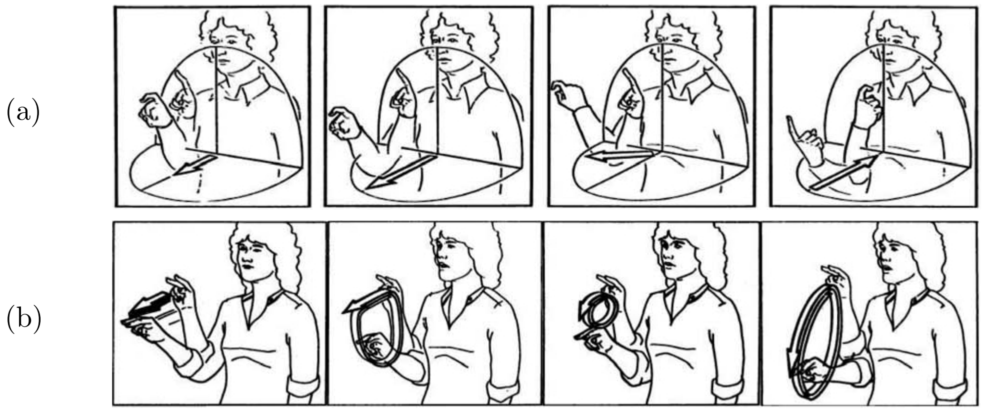


Figure 2.3. Inflection of the sign "ASK": (a) Personal aspects (from left to right): lexical form of "ASK", "I ask you", "I ask her/him", "you ask me" (from [Klima & Bellugi 79]). (b) Temporal aspects (from left to right): "ask regularly", "ask over and over again", "ask continuously", "ask for a long time" (from [Poizner & Klima$^+$ 83])

Incorporation gives the ability to deliver parallel information on different levels. Additional information to the basic sign can be delivered with manual and non-manual components. The upper part of the body, for example, is used to indicate a role change of the speaking character in direct speech. Lips can be used to distinguish between signs with the same manual components. The modification of adjectives and adverbs is also done with non-manual devices. To incorporate several meanings, multiple signs of a phrase can be merged to a single new sign, containing aspects of all separate signs, with the semantic of the whole phrase. Incorporation allows fast speech production. [Bellugi & Fischer 72] found out that, although the production of single signs is significantly slower than the production of spoken words, the duration of signed and spoken sentences with the same content are nearly equal.

2.4 Notation Systems

It is possible to represent sign language in written form, because only a limited set of components is used in sign language.

Notation systems for a description of the syntactic form of a sign are based on a phonological model, which describes the organization of the basic components in subunits. A subunit of sign language was formerly named *chereme* but it has been replaced by the term *phoneme*, which is also used for spoken languages. Currently there is neither an official definition of subunits nor a common pronunciation lexicon for ASL or DGS.

The *Stokoe model* defines 55 phonemes for ASL in three classes: hand configuration, place of articulation and movement (as described Section 2.1). A sign is described with at least one phoneme of each class, whereas the phonemes occur simultaneous. This model does not give a precise definition of a sign, as it is ambiguous in some aspects.

[Liddell & Johnson 89] introduces the *movement-hold model*, where signs are defined as a sequence of movement and hold parts. In movement parts some components change, in hold parts the configuration of the signer stays constant. This model is used by [Vogler & Metaxas 99b] for sign language recognition.

More recent models take into account both the simultaneous and the sequential structure of signs [Sandler 03, Vogler & Metaxas 01].

At the moment no official standard for written sign language exists. Commonly used notation systems are briefly described in the following:

Glosses. Glosses give a semantic representation of sign language instead of describing the syntactic form of the signs. The meaning of a sign is written as the stem form of the corresponding word in spoken language. Transcriptions of incorporations (see Section 2.3) can be added to the base form.

HamNoSys. The Hamburger Notation System [Prillwitz & Leven$^+$ 89] describes the syntactic structure of signs with approximately 150 symbols. The transcription of a sign is done on four levels: hand shape, hand orientation, location, and movement. For non-manual components, especially facial expression, there are only few symbols.

SiGML. The Signing Gesture Markup Language [Elliott & Glauert$^+$ 00] defines an XML data format for the representation and transmission of information about sign language sequences. It is based on HamNoSys with extensions for the representation of non-manual components.

Chapter 3

Sign Language Recognition

Approaches to sign language recognition can be divided into isolated sign recognition and continuous sign language recognition. Isolated sign recognition is a special case of gesture recognition. Gesture recognition systems are typically designed to recognize artificial gestures. The user has to learn these gestures in order to communicate with the system. In sign language the interpersonal and intrapersonal variance between performed gestures is usually higher than between gestures of gesture recognition system, due to their natural usage in speech. In addition, the artifically defined vocabulary, i.e. the set of gestures, for a gesture recognition system is designed for an easy discrimination of the gestures, whereas the vocabulary of a sign language recognition includes gestures that are difficult for the system to disambiguate. Gesture recognition is often used to control an electronic device with a limited number of gestures. Isolated sign recognition deals with larger vocabularies. The use of appearance-based methods for gesture recognition is covered by [Dreuw 05].

This work is concerned with continuous sign language recognition, thus with the recognition of complete sign language sentences. One challenge in continuous sign language recognition is the absence of evident sign boundaries. Start and end of each sign have to be estimated by the recognition system in order to classify the single signs. The estimation of sign boundaries is done implicitly during the recognition process of the presented system. Furthermore, in continuous signing the hands have to move from the ending location of one sign to the starting location of the next. Hand orientation and hand shape change between signs as well. These segments between signs, called *movement epenthesis* [Liddell & Johnson 89], are not part of either of the signs. The appearance of a sign can change depending on the preceeding and the succeeding sign. These *coarticulation effects* are similar to those observed in speech recognition [Hwang & Hon$^+$ 89].

Sign language recognition is considered as a type of speech recognition in this work. The investigated methods have been applied successfully in automatic speech recognition. The following sections give a short overview on the fundamentals of speech recognition in general and the used methods for sign language recognition. A survey of available sign language recognition systems is given in Section 3.6.

3.1 System Overview

Current speech recognition systems use a statistical approach [Jelinek 98]. Given a sequence of features $x_1^T = x_1, \ldots, x_T$ describing the input data, the best sequence of words $w_1^N = w_1, \ldots w_N$ (to simplify matters we refer to a sign as word in the following) is chosen according to Bayes' decision rule which maximizes the *a-posteriori* probabilty [Bayes 63]:

$$\begin{aligned} [w_1^N]_{opt} &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(w_1^N | x_1^T) \right\} & (3.1) \\ &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(w_1^N) \cdot p(x_1^T | w_1^N) \right\} & (3.2) \end{aligned}$$

Equation (3.2) introduces the two basic stochastic models: $p(x_1^T|w_1^N)$ is the probability of observing a sequence of features x_1^T given a word sequence w_1^N. $p(x_1^T|w_1^N)$ is referred to as the *visual model*, according to the acoustic model in speech recognition. The *language model* $p(w_1^N)$ provides the a-priori probability for a word sequence w_1^N.

The basic architecture of a statistical speech recognition system is shown in Figure 3.1. Different parts of the system are described in the following sections. Word models are described in Section 3.2, the language model in Section 3.3. Section 3.5 depicts the search module. The focus of this work is the feature analysis, covered by Chapter 5.

3.2 Visual Modeling

The aim of the visual models is to provide stochastic models of speech units. These models capture static features of speech as well as temporal features.

The acoustic models of speech recognition systems (especially those for large vocabulary speech recognition) use sub-word units such as phonemes to model whole words. These sub-word units are shared among all words in the vocabulary. The model of a whole word is built by connecting the models of the sub-word units according to a pronunciation lexicon. The usage of sub-word units allows to recognize words, for which no example is available in the training phase of the system, by only providing knowledge about the pronunciation of the word. Another advantage of sub-word units is that their models can be estimated more reliable, because more training data is available for each of them.

As mentioned in Section 2.4, no proper sub-word definition for sign language is available. The system presented here uses whole word models as applied, for example, in spoken digit recognition and command-and-control tasks.

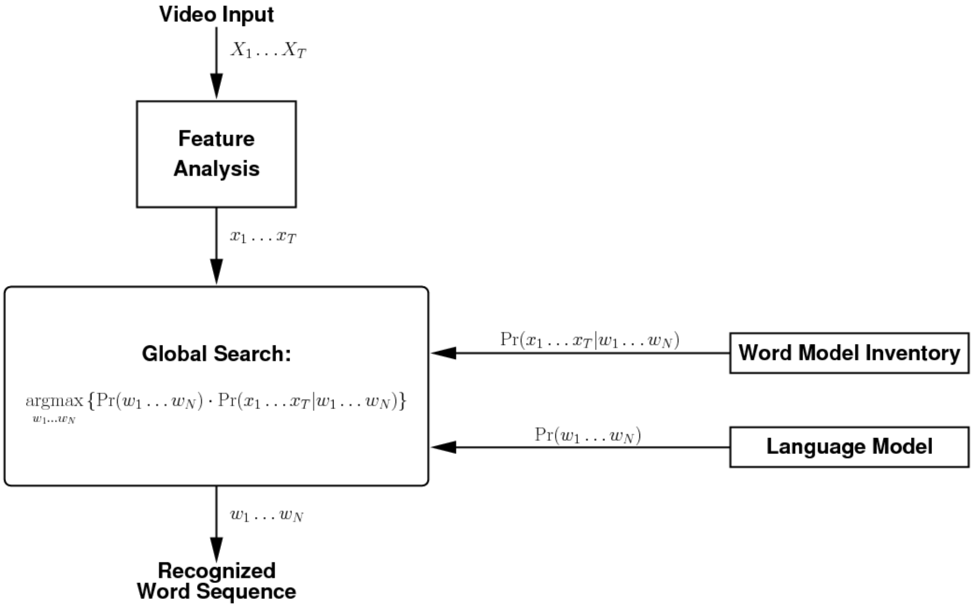


Figure 3.1. Basic architecture of a statistical speech recognition system

3.2.1 Hidden Markov Models

Hidden Markov models (HMM) are used to model the features of speech units. In the presented system, the speech units are whole words. HMMs allow proper modeling of variations in speaking rate. They have been established as the de facto standard in speech recognition [Baker 75, Rabiner 89]. Their ability to model variations of speed and amplitude has been proven also for gesture recognition [Schlenzig & Hunter$^+$ 94, Bobick & Wilson 97, Pavlovic & Sharma$^+$ 97] and sign language recognition [Starner & Pentland 94, Assan & Grobel 97, Vogler & Metaxas 99a] (see also Section 3.6).

HMMs are stochastic finite state automata. Each part of a word is represented by states of the automaton. These states are an abstract concept and cannot be observed, they are *hidden*. The presented system uses the Bakis topology [Bakis 76], where each state has three outgoing transitions: a loop transition to stay in the state, a forward transition to the next state, and a skip transition to the state after the next one (see Figure 3.2). The skip transitions are disabled in some experiments, as discussed in Section 7.2.4.

The probability $p(x_1^T|w)$ of observing the feature sequence x_1^T given a word w is defined as the sum over all possible state sequences s_1^T for this word:

$$p(x_1^T|w) = \sum_{[s_1^T]} p(x_1^T, s_1^T|w) \tag{3.3}$$

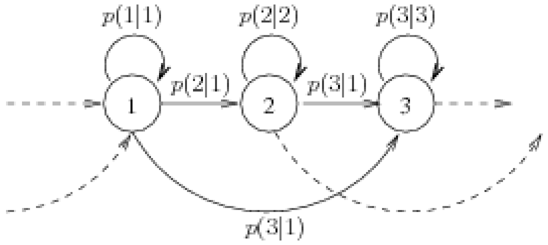


Figure 3.2. Hidden Markov model with 3 states in Bakis topology

$p(x_1^T, s_1^T|w)$ can be rewritten as:

$$p(x_1^T, s_1^T|w) = \prod_{t=1}^{T} p(x_t, s_t|x_1^{t-1}, s_1^{t-1}, w) \tag{3.4}$$

HMMs for subunits, if existent, are concatenated to models for whole words. These word HMMs are concatenated to model a whole word sequence. Using (3.3) and (3.4) one can write the probability of observing x_1^T given a sentence w_1^N as

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t, s_t|x_1^{t-1}, s_1^{t-1}, w_1^N) \ , \tag{3.5}$$

where the sum is taken over all possible state sequences s_1^T for the given word sequence w_1^N. Using Bayes' identity, (3.5) can be rewritten to:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|x_1^{t-1}, s_1^t, w_1^N) \cdot p(s_t|x_1^{t-1}, s_1^{t-1}, w_1^N) \tag{3.6}$$

With the model assumptions that the probability of observing x_t depends only on state s_t and that state s_t depends only on the preceeding state s_{t-1} (*first order Markov assumption*), the equation can be further simplified:

$$p(x_1^T|w_1^N) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \tag{3.7}$$

Equation (3.7) splits $p(x_1^T|w_1^N)$ into a *transition probability* $p(s_t|s_{t-1}, w_1^N)$ and an *emission probability* $p(x_t|s_t, w_1^N)$. The emission probability denotes the probability of observing feature x_t in state s_t. The probability of moving from state s_{t-1} to state s_t is given by the transition probability (see also Figure 3.2).

The sum in (3.7) is approximated by the maximum. This approximation is called the *Viterbi* or *maximum approximation* [Ney 90].

$$p(x_1^T|w_1^N) \approx \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \tag{3.8}$$

Both (3.7) and (3.8) can be evaluated efficiently with the *forward-backward algorithm* [Baum 72, Rabiner & Juang 86] or with *dynamic programming* [Bellmann 57, Viterbi 67, Ney 84].

If the transition probability is assumed to be independent of the word model containing the transition, the transition probability can be replaced by an auxiliary function $q(s_t - s_{t-1})$. In case of the Bakis topology q has to be defined for $q(0)$ (loop-transition), $q(1)$ (forward-transition) and $q(2)$ (skip transition). $q(s_t - s_{t-1})$ is called *time distortion penalty (TDP)*. In the presented system the TDPs are defined by fixed values instead of estimated them on the training data.

$$p(x_1^T|w_1^N) \approx \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot q(s_t - s_{t-1}) \right\} \tag{3.9}$$

3.2.2 Mixture Densities

In the presented system, emission probabilities are modeled with continuous probability distributions. The mixture densities consist of a weighted sum of Gaussian probability densities, as in speech recognition:

$$\begin{aligned} p(x|s, w_1^N) &= \sum_{l=1}^{L_s} p(l|s, w_1^N) \cdot p(x|s, l, w_1^N) && (3.10) \\ &= \sum_{l=1}^{L_s} c_{sl} \cdot \mathcal{N}(x|\mu_{sl}, \Sigma_{sl}, w_1^N) \qquad \text{with} \qquad \sum_{l=1}^{L_s} c_{sl} = 1 && (3.11) \end{aligned}$$

where L_s is the number of densities, c_{sl} denotes the mixture weights and $\mathcal{N}(x|\mu, \Sigma)$ is the normal distribution with mean μ and covariance Σ. For efficiency reasons, a diagonal covariance matrix is used, assuming statistical independence of the features. Furthermore, to avoid problems of estimating the covariance of high dimensional features, the number of parameters to be estimated is reduced by pooling the covariance over all states, such that $\Sigma_{sl} = \Sigma$. For a state s of a single word w the emission probability is:

$$p(x|s, w) = \sum_{l=1}^{L_s} c_{sl} \cdot \mathcal{N}(x|\mu_{sl}, \Sigma, w) \tag{3.12}$$

The sum over all densities is approximated with the maximum over all densities. This approximation is possible, because the mixture probability is dominated by one density in most cases. The advantage of this approximation is a reduced computational complexity in training and in recognition. Using the maximum approximation we get

$$p(x|s,w) \approx \max_{l} \{c_{sl} \cdot \mathcal{N}(x|\mu_{sl},\Sigma,w)\} \tag{3.13}$$

with

$$\mathcal{N}(x|\mu_{sl},\Sigma,w) = \frac{1}{\prod_{d=1}^{D}\sqrt{2\pi\sigma_d^2}} \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_d-\mu_{lswd}}{\sigma_d^2}\right)^2\right) \quad , \tag{3.14}$$

where σ_d is the d-th diagonal entry of the covariance matrix.

The negative logarithm is applied to speed up calculations and for numerical stability reasons:

$$-\log p(x|s,w) = \min_{l}\left\{\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x-\mu_{lswd}}{\sigma_d^2}\right)^2 - \log c_{sl} + \frac{1}{2}\sum_{d=1}^{D}\log(2\pi\sigma_d^2)\right\} \tag{3.15}$$

The negative logarithm $-\log p(x|s,w)$ can be interpreted as a distance between the observed feature x and a reference model μ_{sw}. This distance is called *score* of x, denoted $d(x;s,w)$. A low score of x means, that x fits the model.

3.3 Language Modeling

The purpose of the language model is to provide a model for syntax and semantics of speech. A stochastic model is used which provides an a-priori probability $p(w_1^N)$ of a word sequence w_1^N, which is independent of the visual model. In the field of speech recognition, especially large and very large vocabulary speech recognition, the language model gives a significant contribution to the recognition performance.

The probability of observing a word sequence w_1^N is

$$p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \ . \tag{3.16}$$

Because the number of possible word sequences is unlimited, this probability can not be estimated without some model assumptions. A word sequence is assumed to follow a $(m-1)$-th order Markov process. Thus a word w_n depends only on its $(m-1)$ predecessors $h_n := w_{n-m+1}^{n-1}$, referred to as the *history* of word w_n. The language model probability is then

$$p(w_1^N) = \prod_{n=1}^{N} p(w_n|h_n) \ . \tag{3.17}$$

This model is called a *m-gram language model* [Bahl & Jelinek$^+$ 83]. m-gram language models with a history length of 1, 2 and 3 are called *unigram*, *bigram* and *trigram*, respectively. A trigram language model is used in this work.

For the evaluation of language models the *perplexity* is commonly used. The perplexity of a language model and a test corpus w_1^N is defined as:

$$\begin{aligned} PP &= p(w_1^N)^{-\frac{1}{N}} && (3.18) \\ &= \left[\prod_{n=1}^{N} p(w_n|h_n)\right]^{-\frac{1}{N}} && (3.19) \end{aligned}$$

As the perplexity is an inverse probability, it can be interpreted as the average number of possible words at each position in the text. The logarithm of the perplexity

$$\log PP = -\frac{1}{N}\sum_{n=1}^{N} \log p(w_n|h_n) \qquad (3.20)$$

is equal to the entropy of the text, i.e. the redundancy of words in the test corpus, with respect to this language model.

In equation (3.2), the acoustic model and the language model have the same impact on the decision. Experiments in speech recognition have shown that the recognition performance can be improved, if the language model has more weight than the acoustic model. The weighting is done by introducing a language model scale α and an acoustic model scale β:

$$\begin{aligned} \underset{w_1^N}{\operatorname{argmax}} \left\{p(w_1^N|x_1^T)\right\} &= \underset{w_1^N}{\operatorname{argmax}} \left\{p^{\alpha}(w_1^N) \cdot p^{\beta}(x_1^T|w_1^N)\right\} && (3.21) \\ &= \underset{w_1^N}{\operatorname{argmax}} \left\{\frac{\alpha}{\beta} \log p(w_1^N) + \log p(x_1^T|w_1^N)\right\} && (3.22) \end{aligned}$$

The factor $\frac{\alpha}{\beta}$ is referred to as *language model factor.*

Details about the language model implementation of the presented system can be found in [Wessel & Ortmanns$^+$ 97].

3.4 Training

As described in Sections 3.2 and 3.3, we use stochastic models as knowledge sources of the recognition system. The true model distributions are not known and have to be estimated from training data.

3.4.1 Training of the Visual Model

As described in Section 3.2, Gaussian mixture densities are used for the emission probabilities $p(x|s,w)$. Their parameters mean μ_{lsw}, mixture weights c_{lsw} and variance σ have to be estimated. The parameters are summarized as a parameter set ϑ. A set of R pairs is assumed as training data, each consisting of a feature vector sequence $[x_1^{T_r}]_r$ and a corresponding transcription $[w_1^{N_r}]_r$. The feature vectors are calculated by the feature analysis (see Chapter 5) from the original input data.

The training data is processed sentence wise. The word boundaries are not known and have to be estimated, too. This estimation is done implicitly by constructing a super HMM for each sentence by concatenating the individual word HMMs.

The parameters are estimated with the maximum likelihood principle using an *expectation maximization (EM)* algorithm. The parameter set $\hat{\vartheta}$ is searched which maximizes the likelihood of the training data:

$$\hat{\vartheta} = \underset{\vartheta}{\operatorname{argmax}} \left\{ \prod_{r=1}^{R} p\left([x_1^{T_r}]_r \mid [w_1^{N_r}]_r, \vartheta \right) \right\} \tag{3.23}$$

Because we use the Viterbi approximation (3.8), a feature vector contributes to the estimation of exactly one emission probability $p(x|s,w)$. The training is done with the following algorithm:

1. Estimate the best path, i.e. the best sequence of HMM states. This mapping of feature vectors to HMM states is called *time alignment*.
2. Collect the observations (feature vectors) for each state.
3. Estimate the model parameters for the emission probabilities.

This procedure is iterated until the alignment remains stable or a fixed number of iterations is reached.

The training algorithm needs an initialization, i.e. an initial alignment to estimate initial model parameters. *Linear segmentation* is used for the initialization. In a first step the sequence of feature vectors x_1^T is split into three parts:

$$\underbrace{x_1 \ldots x_{b-1}}_{\text{silence}} \; \underbrace{x_b \ldots x_e}_{\text{speech}} \; \underbrace{x_{e+1} \ldots x_T}_{\text{silence}}$$

The *start-stop detection* searches optimal b and e by minimizing the log likelihood of the segments using two Gaussian densities, one for speech and one for silence [Bridle & Sedgwick 77]. A problem specific to sign language recognition is the selection of the feature which should be used for the detection of silence. This problem is discussed in Chapter 5.

The feature vectors, which are classified as silence, are assigned to a silence model. The feature vectors in the detected speech segment are linearly aligned with the HMM states. Initial model parameters are estimated using the aligned features.

To increase the number of densities in the mixture densities, successive splitting of the mixture densities is applied [Ney 05b, Chapter 2.2].

3.4.2 Training of the Language Model

The transcription of the training data is used to estimate the language model. The training criterion for the language model training is minimal perplexity of the language model on the training data. A closed form solution exists for the maximum likelihood training. The log likelihood is maximized:

$$F = \sum_{h,w} N(h,w) \log p(w,h) \qquad \text{with} \qquad \sum_{w} p(w|h) = 1 \quad \forall h \ , \tag{3.24}$$

where $N(h,w)$ is the number of word sequences $h\ w$ in the training text. The solution of this maximization is:

$$p(w|h) = \frac{N(h,w)}{N(h)} \tag{3.25}$$

The number of possible m-grams grows exponentially with the history size. With the used trigram language model, we get $|V|^3$ possible trigrams for a vocabulary of size $|V|$. Therefore, a large number of m-grams will not be observed in training. An unseen m-gram would have a zero likelihood. To avoid this problem, smoothing techniques are applied. The smoothing is based on discounting, where the probability mass is shifted from seen to unseen events [Katz 87, Ney & Essen$^+$ 94, Generet & Ney$^+$ 95, Ney & Martin$^+$ 97].

The SRILM toolkit [Stolcke 02] is used to estimate the language model parameters.

3.5 Recognition

The aim of the recognition process is to find a word sequence $[w_1^N]_{opt}$ which maximizes the posterior probability given a sequence of feature vectors x_1^T. With the models derived in Section 3.2 and Section 3.3, the task can be formulated as the following optimization problem:

$$\begin{aligned} [w_1^N]_{opt} &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(w_1^N | x_1^T) \right\} \\ &= \underset{w_1^N}{\operatorname{argmax}} \left\{ \left[\prod_{n=1}^{N} p(w_n|h_n) \right] \cdot \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \right\} \end{aligned} \tag{3.26}$$

For a vocabulary consisting of $|V|$ words, the number of possible word sequences of length less or equal to N grows exponentially with the sequence length:

$$\sum_{n=0}^{N} |V|^n = \frac{|V|^{N+1} - 1}{|V| - 1} \tag{3.27}$$

Viterbi-search is used, which solves the problem by dynamic programming [Ney 84]. At each time step the HMM states of all hypotheses are expanded, i.e. the set of successors for each state is calculated. A hypothesis consists of a sequence of words recognized so far up to the current time step and the path of states in the hypothesized current word. The likelihood of all hypotheses can be directly compared and unlikely hypotheses can be discarded, because Viterbi-search is time-synchronous. The discarding of unlikely hypotheses is called *pruning* [Ney & Mergel$^+$ 87].

3.5.1 Evaluation

For the evaluation of recognition results, the *word error rate (WER)* is commonly used. The WER is calculated using the Levenshtein distance [Levenshtein 66], also called *edit distance*, between the true word sequence w_1^N and the recognized word sequence $\hat{w}_1^{\hat{N}}$. This distance is defined as the minimal number of edit operations needed to transform one sequence into the other. Edit operations are *substitutions*, *insertions* and *deletions*.

$$WER = \frac{\#\text{substituions} + \#\text{insertions} + \#\text{deletions}}{\#\text{reference words}} \tag{3.28}$$

The edit distance can be calculated efficiently with a dynamic programming algorithm [Ney 05b].

3.6 State of the Art in Sign Language Recognition

Research in automatic sign language recognition is done for both isolated sign recognition and continuous sign language recognition. It is difficult to compare the available systems, because they differ in many aspects. A main difference is the type of data acquisition: direct measurement or vision-based. Direct measurement is done with data gloves [Fang & Gao 02] or motion capturing systems [Vogler & Metaxas 01], which provide for example 3D spatial information of the hand, fingers, and other body parts with high accuracy and high sampling rates. However, the user is forced to wear a device limiting his movements and needs complex preparation and calibration. These expensive systems can be used mainly in research environments but not in real world applications.

Vision-based systems acquire the used data with video cameras, which are widely used in combination with personal computers. To capture the whole signing space, the entire upper body needs to be in the field of view of the camera. To detect the hands and fingers of the signer, some systems require the signer to wear colored gloves or long sleeved cloth. Many system have constraints regarding lighting, background (uniform colored, static) or the position of the signers. However, many of these constraints conflict with recognizing sign language in a natural context.

A further problem arises from the lack of a common database for the evaluation of sign language recognition systems. Almost all databases used by the different research groups are not publicly available, making a comparison of performance nearly impossible. Additionally, the used databases differ in language, vocabulary size, grammar restriction, and selection of signs [Ong & Ranganath 05].

One should distinguish between signers who learned the gestures only for the recording of a database, and native signers who use sign language in normal communication, because of their different manners of signing. To allow statements about recognition performance in real world applications, the evaluation should be done with recordings of native signers. Most publications do not mention the signers' background in sign language.

Speaker dependence is another important aspect which can be taken into account. Systems which recognize signs for a single signer, who has previously trained the system, are called *speaker dependent.* Speaker independent systems use many signers for training and aim at recognizing signs of arbitrary signers. In speech recognition, speaker dependent systems achieve error rates that are a factor of 2 or 3 lower than those of speaker independent systems [Woodland 01].

An overview about research in sign language recognition, feature extraction and classification methods is given in [Ong & Ranganath 05]. The authors also discuss the analysis of inflection, non-manual components, and grammatical processes in sign language.

This work focuses on vision-based sign language recognition, thus the following two sections present an overview on vision-based systems for isolated sign recognition and continuous sign language recognition.

3.6.1 Recognition of Isolated Signs

One of the first vision-based systems for isolated sign recognition is presented in [Assan & Grobel 97]. The system uses colored gloves for the detection of both hands and the fingers of the dominant hand. Their database consists of 262 signs (sign language of the Netherlands), performed by two non-native signers who learned the signs for this task. The color of the signers' cloth is equal to the background color. A rule-based "classifier" detects the shoulders and the vertical body axis. The feature vector includes hand shape, orientation of the hands and of the fingers of the dominant hand, and the hand position normalized with respect to shoulder and vertical body axis. The HMM classifier achieves error rates between 7 % and 9 % for the person dependent recognition. If videos of both signers have been used in training and test, an error rate of 8 % is achieved.

[Huang & Huang 98] presents a model based tracking method, witch assumes that the hand shape does not change much between consecutive frames. The tracker needs an initialization with difference images which requires the hand to be the only moving object. Additionally, the signer has to be dressed in dark cloth with long sleeves in front of a dark background. The tracked hand region is used for the calculation of the features: global motion, hand orientation and Fourier descriptors for hand shape. Features are only extracted for frames, in which the hand shape changes. Classification is done by a 3D Hopfield neural network with an error rate of 9 % for 15 signs.

In [Holden & Owens 00] colored gloves are required for a 3D hand model based tracker. The used signs have a fixed starting and ending hand posture. The estimated configuration of the hand model is the input of an adaptive fuzzy expert system, which classifies 22 signs (Australian Sign Language mixed with artificial signs) with 5 % error rate. Training and test are performed with one utterance per sign.

A time-delay neural network is used for classification in [Yang & Ahuja$^+$ 02]. The described system uses pixel level motion trajectories of the hand as feature. These trajectories are created with image segmentation and motion segmentation. The hand and head regions are selected and merged using a skin color model and geometric constraints. For 40 ASL signs and an unknown number of samples the system achieves an error rate of 4 %.

[Tanibata & Shimada$^+$ 02] describes a system for the recognition of signs in Japanese Sign Language. In an initialization step the positions of hand, head, and elbows are detected using restrictive person specific templates but allowing complex background. These positions are used for the segmentation of skin, cloth, head, and elbows. Hand and face are distinguished with texture templates from preceeding detections. Geometric features of the hand shape, hand position, and direction of hand motion are extracted from the segmented and labeled hand regions. HMMs are trained for both hands in such a way that each HMM state models either hand shape or motion. The state boundaries are checked manually. 65 samples have been selected manually such that all features are extracted correctly. The authors report no error for the classification of these selected samples.

In [Zhang & Chen$^+$ 04] geometric features for both hands and the fingers of the dominant hand are extracted using pupil detection, colored gloves, background subtraction, and geometry constraints. HMMs are trained with 4 samples of each sign for a vocabulary of 439 signs (Chinese Sign Language). One sample of each sign is used for person dependent recognition, resulting in an error rate of 7 %. The authors carried out additional experiments with tied mixture densities, which yield an error rate of 8 %. The usage of tied mixture densities improves the runtime of the recognition by a factor of 2.

A binary feature vector, called linguistic feature vector, is proposed in [Bowden & Windridge$^+$ 04]. The described system uses a two stage classification. In the first stage position, movement, and shape are classified. These classification results are encoded in a 34 dimensional binary feature vector. In the second stage independent component analysis (ICA) is applied to project the binary feature vector into an Euclidean feature space of lower dimension. The classification is done with Markov chains, which have been trained with only one sample for each of the 49 signs (British Sign Language), resulting in an error rate of 16 %.

[Zieren & Kraiss 05] describes a system that uses features for both hands. Face detection is done using skin color and geometry constrains. The complex background is removed by subtracting the median on pixel level. The hands are tracked with skin color segmentation, a bio-mechanical body model, and other heuristics. Hand positions, geometric features, and their derivatives (22 features in total) are normalized using position and size of the face. The HMM classifier achieves an error rate of 1 % for person dependent classification of 229 signs. The experiments for person independent recognition with four signers (three for training, one for test) yield error rates between 69 % and 56 % depending on the combination of training and test signers. Signer adaptation techniques applied to this system are proposed in [von Agris & Schneider$^+$ 06] which yield an error rate of 21 % for the speaker independent recognition of 153 signs using supervised adaptation.

The person dependent system is extended with non-manual features in [Canzler 05]. Additional features for head position, eye gaze and contour of the mouth are added. Experiments are done on a database of 145 signs. The system achieves an error rate of 36 % with non-manual features only. Using only manual features, the systems achieves an error rate of 3 %. The combination of both feature types improves the result obtained with manual features only by 0.22 %.

The usage of self organizing subunits is analyzed in [Bauer & Kraiss 02]. The subunits are defined data-driven without linguistic models. Features regarding size, shape, and position of hands, fingers and body are extracted using colored gloves. The system achieves an error rate of 7 % on 100 signs with 150 subunits (7 samples per sign in training, 3 samples in test). 50 new signs, automatically transcribed using the trained subunit models, are recognized with an error rate of 19 %.

3.6.2 Continuous Sign Language Recognition

One of the first vision-based system for continuous sign language recognition is described in [Starner & Pentland 94]. A segmentation based on colored gloves is used to extract features for hand position, angle, and hand shape. The system is tested with 40 ASL signs in sentences consisting of 5 signs each, constructed with a fixed grammar (pronoun, verb, noun, adjective, same pronoun). 395 sentences are used to train HMMs. The person dependent recognition of 99 sentences achieves an word error rate of 1 % with the fixed grammar and 9 % without grammar restrictions.

[Starner & Weaver$^+$ 98] describes an extended system, where no colored gloves are necessary, because hands are detected with skin color segmentation. The authors have carried out two experiments with different types of videos: one showing the frontal view of a signer in front of a complex background, the second video is recorded by a hat-mounted camera. The hat-mounted camera is pointed downward towards the hands and showes mainly the hands. Absolute position, movement, and shape features are used for classification. For the evaluation, 384 sentences are used in training, and 94 are used for testing. The sentences have the same restrictions as in [Starner & Pentland 94]. A word error rate of 8 % is achieved with the frontal view videos. The videos captured by the hat-mounted camera have been recorded with another signer and are recognized with a word error rate of 2 %.

In [Holden & Lee$^+$ 05] the authors describe a tracking algorithm, which uses skin color detection, and a so called correspondence algorithm that identifies hands and face. Position, shape and movements of the hands are used as features. The HMM classifier achieves an error rate of 3 % on sentence level and a word error rate of 1 %. The 14 distinct sentences consist of 21 signs (Australian Sign Language) and are constructed with a fixed grammar, which is also used by the recognition system. 216 of 379 utterances are used for training, but each sentence occurred both in the training set and in the test set.

The subunit approach presented in [Bauer & Kraiss 02] is also applied to continuous sign language recognition in [Bauer 04] using the same features. The subunits are created based on isolated signs. Experiments with 52 signs in 100 distinct sentences result in a word error rate of 12 %.

Chapter 4

Tracking

The aim of tracking methods is to detect and to follow one or several objects in a sequence of images. It can be seen as a kind of object detection in a series of similar images. In most cases, tracking methods are applied to videos, which means a large number of images has to be processed. Therefore, methods used in object detection tasks are normally not applicable, because they need too much computation time. Many applications require processing in real-time. Furthermore, the knowledge about previous object positions can be used to predict and detect objects in following images.

Appearance-based features (see Section 5.1) often use a downscaled input image. Details of small image regions, like the hand and the face of the signer, are not visible in the scaled image. Tracking gives the possibility to extract specific regions of interest out of the image. Not only the extracted image patch can be used as feature, but also the detected object positions. In the field of sign language recognition tracking methods can be used to keep track of the hands and of the head. The tracked head and hands are used for the computation of visual features and positional features (see Chapter 5).

Section 4.1 gives an overview about state of the art methods for tracking. The approach used in this work is introduced in Section 4.2. The applications to hand tracking and head tracking are shown in Section 4.3 and Section 4.4.

4.1 State of the Art

Tracking methods are applied to many different tasks including gesture recognition, human movement tracking, face recognition, aerial surveillance and traffic supervision. A good overview on tracking methods used in gesture and human movement recognition can be found in [Gavrila 99]. Some of the widely used methods are described briefly in this section.

[Comaniciu & Ramesh^{+} 00] presents the Meanshift algorithm which tracks non-rigid objects based on visual features such as color and texture. Statistical distributions are used to characterize the object of interest. The algorithm tolerates partial occlusions of the tracked object, clutters, rotation in depth and changes in camera position.

The Camshift (continuously adaptive meanshift) algorithm is a extension of the Meanshift algorithm that is able to deal with dynamically changing color probability distributions. In the system presented in [Bradski 98] it is used to track human faces in real-time.

In [Isard & Blake 98] the Condensation (conditional density propagation) algorithm is presented. It is a model based method that is able to track objects in visual cluttered scenes.

An earlier version of the tracking method described in this work and its application to sign language recognition is described in [Dreuw & Deselaers$^+$ 06].

4.2 Tracking using Dynamic Programming

The task of tracking one object in an image sequence $X_1^T = X_1, \ldots, X_T$ can be formulated as an optimization problem. Expressed in a probabilistic framework, the path of object positions $u_1^T = u_1, \ldots, u_T$ is searched that maximizes the likelihood of this path given the image sequence X_1^T:

$$\begin{aligned}
[u_1^T]_{opt} &= \operatorname*{argmax}_{u_1^T} \left\{ p(u_1^T | X_1^T) \right\} && (4.1)\\
&= \operatorname*{argmax}_{u_1^T} \left\{ \prod_{t=1}^{T} p(u_t | u_1^{t-1}, X_1^t) \right\} && (4.2)
\end{aligned}$$

The advantage of this approach is the optimization over the complete path, which avoids local decisions that might not be correct.

Assuming a first-order Markov process for the path, meaning that an object position depends only the previous position, (4.2) can be simplified to:

$$[u_1^T]_{opt} = \operatorname*{argmax}_{u_1^T} \left\{ \prod_{t=1}^{T} p(u_t | u_{t-1}, X_{t-1}^t) \right\} \tag{4.3}$$

This assumption allows easier modeling of the object behavior, because only succeeding object positions have to be rated. Applying the logarithm to (4.3) yields:

$$[u_1^T]_{opt} = \operatorname*{argmax}_{u_1^T} \left\{ \sum_{t=1}^{T} \log p(u_t | u_{t-1}, X_{t-1}^t) \right\} \tag{4.4}$$

The probability $p(u_t | u_{t-1}, X_{t-1}^t)$ can be expressed by a function $\tilde{q}(u_{t-1}, u_t; X_{t-1}^t)$ that rates the object position u_t with a score depending on the previous position u_{t-1} and the images X_{t-1}^t. In order to fulfill the requirements of a probability density function, the score has to be normalized by the sum over the scores of all possible object positions. The logarithm can be omitted due to its monotonicity:

$$\begin{aligned}
[u_1^T]_{opt} &= \operatorname*{argmax}_{u_1^T} \left\{ \sum_{t=1}^{T} \log \frac{\tilde{q}(u_{t-1}, u_t, X_{t-1}^t)}{\sum_{u'} \tilde{q}(u_{t-1}, u'; X_{t-1}^t)} \right\} && (4.5) \\
&= \operatorname*{argmax}_{u_1^T} \left\{ \sum_{t=1}^{T} \frac{\tilde{q}(u_{t-1}, u_t, X_{t-1}^t)}{\sum_{u'} \tilde{q}(u_{t-1}, u'; X_{t-1}^t)} \right\} && (4.6)
\end{aligned}$$

The normalizing part is constant with respect to an object position u_t, thus it can be omitted for the maximization:

$$[u_1^T]_{opt} = \operatorname*{argmax}_{u_1^T} \left\{ \sum_{t=1}^{T} \tilde{q}(u_{t-1}, u_t; X_{t-1}^t) \right\} \tag{4.7}$$

The score function $\tilde{q}(u_{t-1}, u_t; X_{t-1}^t)$ is split into a function $q(u_{t-1}, u_t; X_{t-1}^t)$ depending on the image sequence, and an image independent part $\mathcal{T}(u_{t-1}, u_t)$ to control properties of the path.

$\mathcal{T}(u, u')$ is function similar to the time distortion penalty of the visual model (see (3.9), page 13). Here one wants to penalize large distances between consecutive object positions, as it is not likely that succeeding object positions are far away from each other. This function is called *jump penalty.* The squared Euclidean distance is used here as the jump penalty:

$$\mathcal{T}(u, u') = \alpha_{\mathcal{T}} \cdot \|u - u'\|^2 = \alpha_{\mathcal{T}} \cdot (u - u')^T \cdot (u - u') \ , \tag{4.8}$$

where $\alpha_{\mathcal{T}}$ is weighting factor for the jump penalty.

Using these score functions, (4.7) can be rewritten to (4.9). The jump penalty $\mathcal{T}(u_{t-1}, u_t)$ is subtracted from the score to penalize wide movements.

$$[u_1^T]_{opt} = \operatorname*{argmax}_{u_1^T} \left\{ \sum_{t=1}^{T} q(u_{t-1}, u_t; X_{t-1}^t) - \mathcal{T}(u_{t-1}, u_t) \right\} \tag{4.9}$$

This optimization problem is solved using dynamic programming. An auxiliary function $C(t, u)$ is introduced, which gives the best score for the path at time t ending in position u.

$$C(t, u) = \max_{u_1^t \,:\, u_t = u} \left\{ \sum_{t'=1}^{t} q(u_{t'-1}, u_{t'}; X_{t'-1}^{t'}) - \mathcal{T}(u_{t'-1}, u_{t'}) \right\} \tag{4.10}$$

$C(t, u)$ can be defined recursively. The auxiliary function needs to be maximized only over the direct predecessor positions:

$$C(t, u) = \max_{u'} \left\{ C(t-1, u') - \mathcal{T}(u', u) + q(u', u; X_{t-1}^t) \right\} \tag{4.11}$$

The maximization does not need to consider all predecessor positions of position u, but a limited set of predecessor positions $\mathcal{M}(u)$. This limitation avoids large distances between consecutive object positions (additional to the jump penalty) and decreases computation time.

The score $C(t,u)$ is calculated for each time step t and each position u successively, starting from $t = 1$, yielding a table of scores. To reconstruct the best path, a table of back-pointers $B(t,u)$ is needed, which stores the best predecessor position for each time step t and each position u:

$$\begin{aligned} C(t,u) &= \max_{u' \in \mathcal{M}(u)} \left\{ C(t-1,u') - \mathcal{T}(u',u) + q(u',u;X_{t-1}^{t}) \right\} && (4.12) \\ B(t,u) &= \operatorname*{argmax}_{u' \in \mathcal{M}(u)} \left\{ C(t-1,u') - \mathcal{T}(u',u) + q(u',u;X_{t-1}^{t}) \right\} && (4.13) \end{aligned}$$

The best path is traced back as follows:

1. Search best last position:

$$u_T = \operatorname*{argmax}_{u} \{C(T,u)\}$$

2. Repeat for $t = T - 1$ down to $t = 1$:

$$u_t = B(t+1, u_{t+1})$$

One can expect that the scores will be high in a region around the true object position. If computation time needs to be decreased, not each possible position has to be evaluated. From $t = 2$ onwards, only those scores have to be calculated, for which the score of the predecessor position u is high enough:

$$C(t-1,u) > \max_{u'} \left\{ C(t-1,u') \right\} - d_0 \qquad (4.14)$$

where d_0 is an accurate pruning threshold. If the pruning threshold is too low, the best path is possibly not found.

This work considers only tracking of image regions of a constant size. The extension to tracking with variable size is described in [Dreuw 05].

4.3 Hand Tracking

The dominant hand in an image sequence is tracked in order to extract manual features for sign language recognition. The dominant hand can be expected to be the object that moves more than every other object in the sequence. Another attribute of the hand is its color. A skin color likelihood can be calculated for color images [Jones & Rehg 02]. Brightness can be used to detect skin in gray level images.

Challenges for hand tracking in sign language videos are:

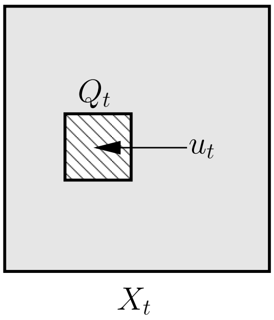


Figure 4.1. Notation used in scoring functions

- Two hands appear in the video. The tracker should not confuse left and right hand during the sequence.
- The hands may overlap. In many signs performed with both hands, the hands overlap at some time. This overlap does not have to be a direct contact of the hands, because of the two dimensional projection of the camera. The same hand should be tracked before and after the crossing of the hands.
- Depending on the camera location, a hand can temporarily disappear. This is mostly the case at sentence begin and sentence end, when the hand is in an idle position. The tracker should find the hand immediately when it appears.
- Many signs are performed in front of the face. Due to the similar color of the face and the hand it is difficult to distinguish between them.

The following section describes different score functions that can be used for hand tracking and in general for the tracking of moving objects changing moderately in appearance.

4.3.1 Score Functions

In the equations of the following sections

$$Q_t := \{u_t + u : u \in Q\} \qquad \text{with} \qquad Q := \{(i,j) : -w \leq i \leq w,\ -h \leq j \leq h\} \quad (4.15)$$

denotes the set of positions in a rectangle of size $w \times h$ around position u_t (see also Figure 4.1). This rectangle is called *tracking window*. $X[u]$ denotes the pixel in image X at position u.

One of the simplest approaches to hand tracking is to search the regions in all images where the most motion occurs. Motion means a difference in consecutive images:

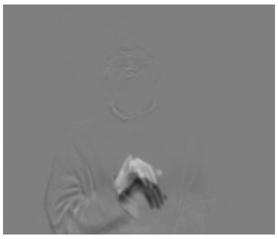


Figure 4.2. A difference image. Pixel values range between -1 (black) and 1 (white)

$X'_t := X_t - X_{t-1}$. An example of such a difference image is shown in Figure 4.2. Moving objects are visible in difference images, while all constant parts of the image have an uniform color. Score functions using only difference images are:

- Motion score

$$q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \in Q_t} X'_t[u] \tag{4.16}$$

- Absolute motion score

$$q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \in Q_t} \left| X'_t[u] \right| \tag{4.17}$$

- Squared motion score

$$q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \in Q_t} \left(X'_t[u] \right)^2 \tag{4.18}$$

When the hand is moving, the image changes at two locations: at the former position and at the new position of the hand. This is considered in the following score function (4.19), by maximizing the motion in successive hypothesized object regions. This means that the regions including the pixels with the highest absolute pixel values in the difference image are detected. In Figure 4.2 these regions are visible as nearly black region (pixel values approximately -1) at the former position and as nearly white region (pixel values approximately 1) at the new position of the hand.

$$q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \in Q_{t-1}} \left(X'_t[u] \right)^2 + \sum_{u \in Q_t} \left(X'_t[u] \right)^2 \tag{4.19}$$

Another approach is to assume that the tracked object is nearly constant in appearance from one image to the next, which means a small distance between two consecutive object appearances. Because (4.13) uses a maximization, the negative distance is used as score:

$$-q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \in Q} (X_t[u_t + u] - X_{t-1}[u_{t-1} + u])^2 \tag{4.20}$$

If this score function is used for tracking, the scores $C(t = 1, u)$ need to be initialized with another score function. Without an initialization, the tracker would stay on some constant background image part. An initialization can be done, for example, with a score for each image region provided by an object detection method.

The score function (4.20) uses the Euclidean distance between two image patches as distance measurement. The Euclidean distance does not account for image transformations such as scaling, rotation, and translation. The tangent distance (introduced by [Simard & LeCun$^+$ 98]) as described in [Keysers & Macherey$^+$ 01] is one approach to incorporate invariance with respect to certain transformations in the distance between two images. Invariance in this context means that image transformations should not have a large impact on the distance between two images. A detailed description of the tangent distance can be found in [Keysers & Macherey$^+$ 04], its application to gesture recognition is presented in [Dreuw & Keysers$^+$ 06].

No initialization is needed, if one assumes a constant background as the tracking criterion. That is, the only image parts assumed to change are those where the hand has been in the previous image and where the hand is in the current image. Thus, the difference in succeeding images between pixels belonging to the background should be minimal. Background pixels are those pixels which do not belong to the hypothesized tracking regions Q_t and Q_{t-1}:

$$-q(u_{t-1}, u_t; X_{t-1}^t) = \sum_{u \notin Q_t \cup Q_{t-1}} \left(X_t'[u]\right)^2 \tag{4.21}$$

The sum over $u \notin Q_t \cup Q_{t-1}$ can be split in components (see also Figure 4.3):

$$\begin{aligned}
\sum_{u \notin Q_t \cup Q_{t-1}} \left(X_t'[u]\right)^2 &= \sum_{u \in U} \left(X_t'[u]\right)^2 - \sum_{u \in Q_t} \left(X_t'[u]\right)^2 - \sum_{u \in Q_{t-1}} \left(X_t'[u]\right)^2 + \sum_{u \in Q_{t-1} \cap Q_t} \left(X_t'[u]\right)^2 \\
&= \mathrm{const}(X_t') - \sum_{u \in Q_t} \left(X_t'[u]\right)^2 - \sum_{u \in Q_{t-1}} \left(X_t'[u]\right)^2 + \sum_{u \in Q_{t-1} \cap Q_t} \left(X_t'[u]\right)^2
\end{aligned} \tag{4.22}$$

The constant part does not need to be considered in the score function, because it is independent of the object position. Using this split into components, it can be seen that score function (4.21) is equal to the total motion score function (4.19) except for the factor that compensates overlapping regions.

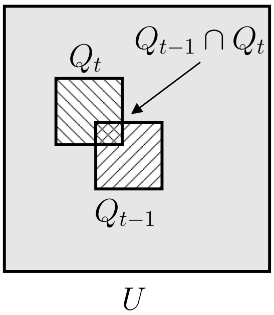


Figure 4.3. Two overlapping image regions

Score functions can be combined to new score functions, to take advantage and to compensate for disadvantages of the single functions. The influence of each functions is weighted with factors α_i

$$q(u_{t-1}, u_t; X_{t-1}^t) = \sum_i \alpha_i \cdot q_i(u_{t-1}, u_t; X_{t-1}^t) \tag{4.23}$$

The score functions (4.20) and (4.21) can be combined. This combination results in a score function which checks that both background and object appearance stay nearly constant. This score function needs no initialization. The influence of both parts is controlled with the weighting factors α_i.

$$-q(u_{t-1}, u_t; X_{t-1}^t) = \alpha_1 \sum_{u \in Q} (X_t[u_t + u] - X_{t-1}[u_{t-1} + u])^2 + \alpha_2 \sum_{u \in Q_{t-1} \cap Q_t} \left(X_t'[u]\right)^2 \tag{4.24}$$

Using (4.22), (4.24) can be rewritten to:

$$\begin{aligned} -q(u_{t-1}, u_t; X_{t-1}^t) \quad = \quad & \alpha_1 \quad \Big(\sum_{u \in Q} (X_t[u_t + u] - X_{t-1}[u_{t-1} + u])^2 \Big) \\ - & \alpha_2 \quad \Big(\sum_{u \in Q_t} \left(X_t'[u]\right)^2 + \sum_{u \in Q_{t-1}} \left(X_t'[u]\right)^2 - \sum_{u \in Q_{t-1} \cap Q_t} \left(X_t'[u]\right)^2 \Big) \end{aligned} \tag{4.25}$$

4.3.2 Integral Images

Most scoring functions presented in Section 4.3.1 use the sum of pixel values in a rectangular region of an image or a difference image. The computation of the sum over a rectangle of size $w \times h$ needs to access $w \cdot h$ pixel values. This computation can be done more efficient using *integral images* [Viola & Jones 04].

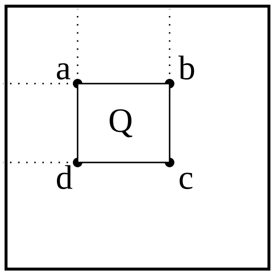


Figure 4.4. The sum of pixels in rectangle Q can be computed using the integral image at locations a, b, c and d

An integral image I is an intermediate representation for an image X, which contains at position (i, j) the sum of pixels above and left of (i, j):

$$I[i,j] = \sum_{i' \leq i, j' \leq j} X[i', j'] \tag{4.26}$$

This can be computed in one pass over the image using the cumulative row sum $S[i, j]$:

$$\begin{aligned} S[i,j] &= S[i,j-1] + X[i,j] & (4.27) \\ I[i,j] &= I[i-1,j] + S[i,j] & (4.28) \end{aligned}$$

The sum of a rectangular image region can now be computed using only four references to the integral image. Using the notation of Figure 4.4, the sum in region Q is:

$$\sum_{u \in Q} X[u] = I[a] + I[c] - (I[b] + I[d]) \tag{4.29}$$

4.4 Head Tracking

The score functions used for hand tracking cannot be used for head tracking, because the head is assumed to move not much in sign language videos.

The simplest approach is to use the property of the head as being the largest skin colored object in the image. In color images the above mentioned skin color probability can be used, if no other objects with skin like color, an orange t-shirt for example, occur in the images. In gray level images the head can be expected to be the brightest region of the image. This works only if no larger skin colored or bright regions appear in the image. Furthermore, this method will not always find the same face region, i.e. the tracking window is not centered on a reference point like the nose.

A face detection method is used in the presented system to track the head of the signer. Widely used face detection methods include:

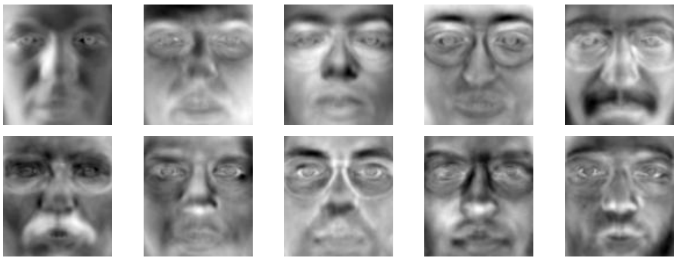


Figure 4.5. The first 10 eigenfaces, calculated on the BioID database

- [Viola & Jones 04] presents a face detection method, which is based on the AdaBoost algorithm from machine learning.
- [Rowley & Baluja$^+$ 98] describes a neural network for face detection.
- The eigenface approach presented in [Turk & Pentland 91] applies principal component analysis to face detection.

The eigenface approach is used for head tracking, because it yields good results and it is easy to integrate into the presented tracking framework. In contrast to the Viola & Jones method, which calculates classifiers that discriminate only between "face" and "no face", the eigenface method calculates distances, that can be used in a score function.

Eigenfaces are calculated by performing principal component analysis (PCA, see also Section 5.1.1) on a training set of face images. The resulting eigenvectors span a subspace, called the *face space*, of the image space. Figure 4.5 shows eigenfaces, which have been calculated on the BioID database[1]. The images of the BioID database show one person per image and are annotated with the face positions.

An image X can be projected to face space by a linear transformation ϕ:

$$\phi(X) = V^T(X - \mu) \tag{4.30}$$

where $V = [v_1 \dots v_m]$ is the matrix of the first m eigenvectors and μ is the mean face calculated on the set of training images. The image X is used in vector form, by sorting the pixel in lexical order. The projection from the face space back to the image space is:

$$\phi^{-1}(X_f) = VX_f + \mu \ , \tag{4.31}$$

where X_f is the image representation in face space $\phi(X)$.

[1] `http://www.humanscan.de/support/downloads/facedb.php`

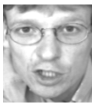
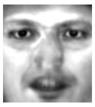
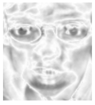
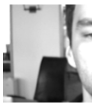
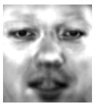
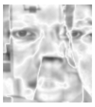
X	$\phi^{-1}(\phi(X))$	$X - \phi^{-1}(\phi(X))$	$d_f(X)$
			278
			432

Figure 4.6. Projected images and their face space distance. The lower non-face image yields a higher face space distance than the face image.

The distance $d_f(X)$ between an image and its forward and backward projected version, is called the *face space distance*. It can be used as a measure of "faceness".

$$d_f(X) = \|X - \phi^{-1}(\phi(X))\|^2 \tag{4.32}$$

An example of projected images and the resulting distance is shown in Figure 4.6.

For face detection the face space distance is calculated for each region of an image. A face is detected at the region with the lowest face space distance. No face is detected, if all distances is are higher than some threshold. The Viola & Jones method fails directly when a hand occludes the face, whereas eigenfaces might still have a low face space distance.

We use the negative face space distance as score function to detect and track heads:

$$q(u_{t-1}, u_t; X_{t-1}^t) = -d_f\left(X_t(u_t)\right) \tag{4.33}$$

where $X_t(u_t)$ denotes a rectangular patch of image X_t centered in position u_t. This score function is combined with a skin color or brightness score function to avoid high scores for face like structures in the background [Dreuw & Deselaers+ 06]. The calculation of the face space distance is relatively time consuming, but the eigenface score function needs to be evaluated only for a few regions of each image if pruning is used. The eigenface approach is not only applicable to face detection but also for the detection of other objects that can be modeled with an eigenspace representation.

Chapter 5

Features for Sign Language Recognition

Most research groups use complex methods to extract features for the recognition of sign language or gestures. Hand shape, hand orientation, finger orientation, or 3D models are often used features. In most cases, the calculation of these features depends on a segmentation of the input image, geometric constraints, and other heuristics. A disadvantage of some of these methods is that local decisions are taken. Especially segmentation of images is difficult and error-prone. A possible segmentation error and errors of a mismatched model or constraint are propagated through the whole system. In contrast, appearance-based approaches do not rely on models for single objects and use the image itself as a feature for recognition.

As mentioned in Section 2.1, signs consist of four basic manual components. The movement of the hands is not observable in single images. Therefore, special methods have to be applied to extract features for this component as well. The presented appearance-based system uses downscaled versions of the input images. The resolution of these images is too low to observe details of special regions of interest, like the signer's hand or face. Therefore, tracking of the signer's hand and head is used to extract appearance-based features for these specific regions. Additionally, features describing the positions of the dominant hand and its movement are investigated.

Appearance-based features of whole images are presented in Section 5.1. Features for manual and non-manual components of sign language are discussed in Section 5.2 and 5.3. The combination of features is described in Section 5.4.

5.1 Image Features

When using appearance-based methods, a gray level image of size $I \times J$ is often represented by a vector in an $I \cdot J$ dimensional vector space. In practice, such a high dimensional vector space is too large to allow robust classification. A common way to resolve this problem is to apply dimension reduction techniques [Martinez & Kak 01]. Principal component analysis and linear discriminant analysis are widely used dimension reduction methods. Linear discriminant analysis is often applied in speech recognition for feature combination and feature reduction [Haeb-Umbach &

Ney 92]. Principal component analysis is used for a wide range of image processing tasks.

5.1.1 Principal Component Analysis

Principal Component Analysis (PCA), also known as Karhunen-Loéve transformation, is an unsupervised approach to select features for a data set. It is a linear transformation that projects a feature vector $x \in \mathbb{R}^D$ to a representation of lower dimension $\hat{x} \in \mathbb{R}^d$. After calculating the eigenvectors of the empirical covariance matrix for the full data set, the eigenvectors are sorted according to their corresponding eigenvalues in decreasing order. The first d eigenvectors are used as columns for the transformation matrix V. This matrix is then used to project a vector to the subspace spanned by the d principal components. It can be shown that this transformation is the best linear transformation to a feature space of dimension d with respect to the representation error $E_x \left\{ \|x - \hat{x}\|^2 \right\}$ [Duda & Hart$^+$ 01, Chapter 3.8.1]. Details about PCA and its applications can be found in [Jolliffe 02].

PCA is expected to capture the most relevant parts of a vector, because it discards directions of low variance. This property makes PCA well suited for the analysis of video frames. If the background of all images does not change much, pixels belonging to the background have a small variance and will be discarded by the PCA. The impact of each pixel on the lower dimensional representation can be visualized by the row sum of V as an image X_V: $X_V[i] = \sum_{d'=1}^{d} |V_{i,d'}|$. Figure 5.1 shows some example images of the data set, the variance and the visualization of the PCA transformation matrix.

PCA does not take into account class information. Thus, it is possible that the discriminative components of the features will be discarded. The dimension reduction of PCA is optimized for an efficient representation of the features, not for class discrimination.

5.1.2 Linear Discriminant Analysis

In contrast to PCA, the aim of the linear discriminant analysis (LDA), also called Fisher's LDA, is to maximize the separability of classes in the transformed feature space. It creates a linear combination of independent features which maximizes the distances between the means of all classes. Details about LDA calculation can be found in [Duda & Hart$^+$ 01, Chapter 3.8.2].

LDA requires class information of the feature vectors in the training set. Thus, in speech recognition one needs to classify the feature vectors, before the LDA can be calculated. Therefore, an initial alignment, i.e. a mapping of feature vectors to HMM states, is done using untransformed features. This alignment is estimated and may contain wrong class information. In other applications of LDA, e.g. in visual object

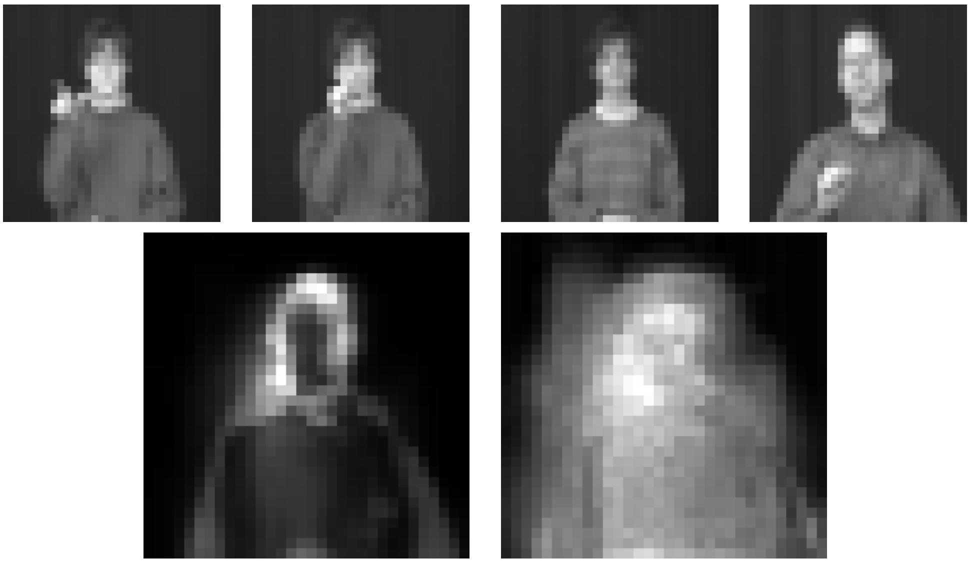


Figure 5.1. First row: example images of the data set ($D = 32 \cdot 32 = 1024$), second row: variance of the data set, visualization of the PCA transformation matrix ($d = 110$). A black pixel means that this component is discarded by the PCA.

classification tasks, the class information of the training samples is usually known.

Scatter matrices, which measure variance in classes and between classes, are used for the calculation of the LDA. Their robustness depends on the dimensionality of the feature vectors and on the number of available feature vectors for each class. Though LDA is theoretically better suited for pattern recognition tasks than PCA, PCA is reported to be more stable for high dimensional data and small training sets [Martinez & Kak 01].

5.1.3 Motion

The silence detection in the linear segmentation step of the training procedure (see Section 3.4.1) needs a feature that can be used to discriminate between silence and non-silence segments. Speech recognition systems often use the energy of the acoustic input signal as the feature for silence detection.

It is not obvious how silence in sign language should be defined. The approach used in the presented system is to define silence as segments without motion. The motion occurring at time t can be measured as *motion energy*, which is defined by the square root of the sum of pixel values in the difference image of two consecutive

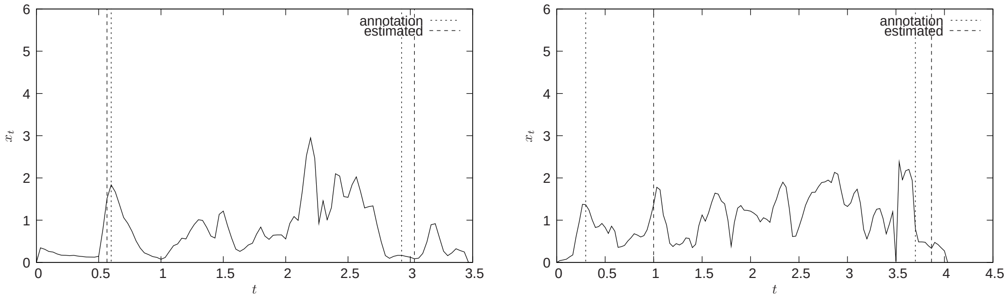


Figure 5.2. Motion energy over time for two example sequences. The annotated begin and end times of the sentence and their estimates are marked.

frames X_{t-1}, X_t:

$$e_t = \sqrt{\sum_{u \in U} (X_t[u] - X_{t-1}[u])^2} \tag{5.1}$$

However, the signer does not move his hands all the time while signing. Some signs include segments where the signer's configuration stays constant for a few frames. A plot of the motion energy over time for two sentences is shown in Figure 5.2. It can be seen that the motion energy is very low in some segments. Furthermore, motion occurs in some sentences also before and after the sentence.

The motion energy feature is used for silence detection in the presented system. Additionally, the use of motion energy as feature for sign language recognition is investigated.

5.2 Manual Features

Manual features describe manual components of a sign (see Section 2.1). In this work, only features for the dominant hand are used. It is assumed that the features of the dominant hand are significant enough for most signs. The positions of the dominant hand are detected with the tracking method presented in Section 4.3.

5.2.1 Hand Position and Velocity

Using only the hand position (see also Figure 5.3), one can compute the following simple features:

- Position of the hand in the two dimensional projection of the signing space: u_t
- Hand motion or velocity: $m_t = u_t - u_{t-\Delta}$, $\Delta \in \{1, 2, \dots\}$

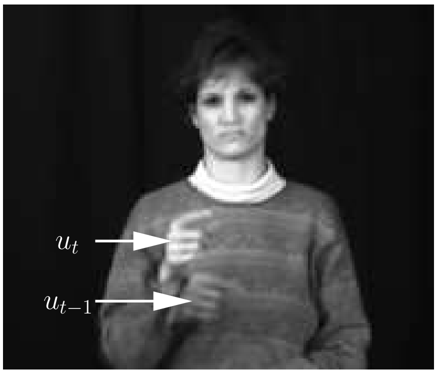


Figure 5.3. Overlay of two consecutive frames with labeled hand positions

5.2.2 Hand Trajectory

Hand position and hand motion describe manual features only in the immediate vicinity of a specific point of time, thus they are local features. The trajectory of the hand describes properties of a sign on a more global level. The global features used in this work are similar to the features presented in [Vogler & Metaxas 99a]. The system of Vogler and Metaxas used 3D hand position information from a motion capturing system.

In order to calculate global features describing geometric properties of the hand trajectory, the covariance matrix for hand positions in a certain time window is estimated. For a window size of $2\Delta + 1$, the covariance matrix Σ_t and its eigenvectors $v_{t,i}$ and eigenvalues $\lambda_{t,i}$ at time t are calculated as follows:

$$\mu_t = \frac{1}{2\Delta + 1} \sum_{t'=t-\Delta}^{t+\Delta} u_{t'} \tag{5.2}$$

$$\Sigma_t = \frac{1}{2\Delta + 1} \sum_{t'=t-\Delta}^{t+\Delta} (u_{t'} - \mu_t)\,(u_{t'} - \mu_t)^T \tag{5.3}$$

$$\Sigma_t \cdot v_{t,i} = \lambda_{t,i} \cdot v_{t,i} \qquad i \in \{1, 2\} \tag{5.4}$$

The eigenvalues $\lambda_{t,i}$ and eigenvectors $v_{t,i}$ of the covariance matrix can then be used as global features. The eigenvector with the larger corresponding eigenvalue can be interpreted as the main direction of movement. The eigenvalues describe the form of the movement. If one eigenvalue is significantly larger than the other, the movement fits a straight line. Figure 5.4 shows examples of trajectories and their eigenvectors and eigenvalues. The eigenvalues can be normalized such that $\sum_i \lambda_{t,i} = 1$. The ratio $\frac{\lambda_1}{\lambda_2}$ can also be used as feature describing the form of the movement.

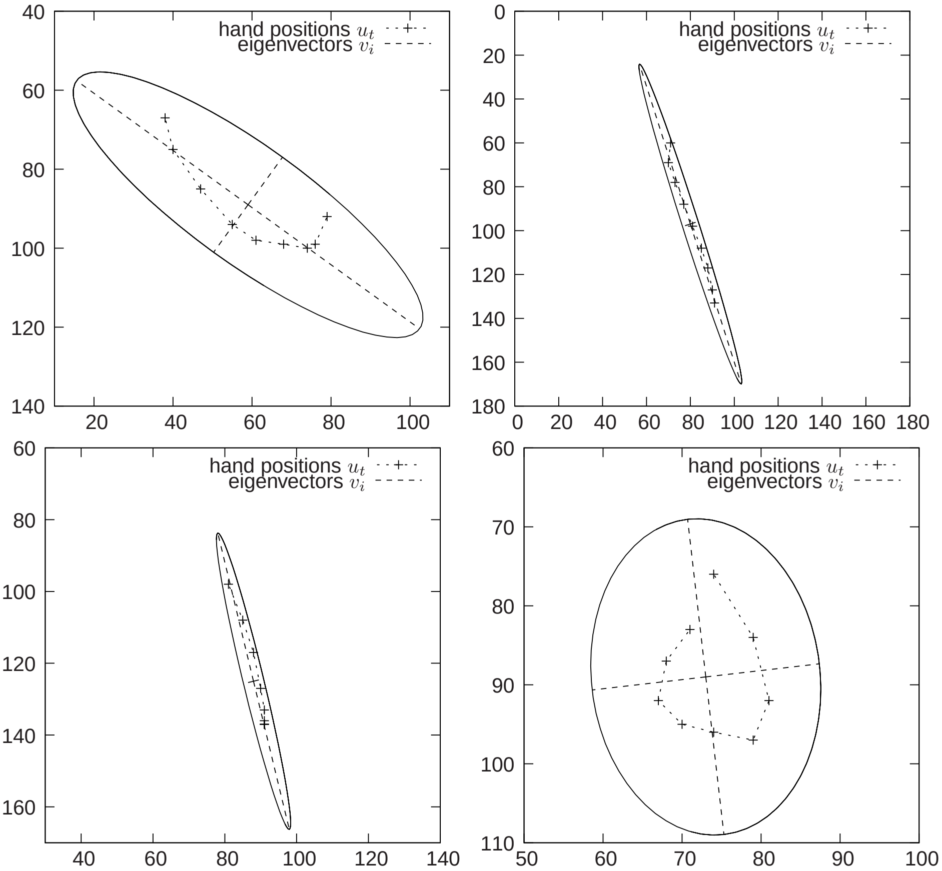


Figure 5.4. Examples of different hand trajectories and corresponding eigenvectors. The covariance matrices are visualized as ellipses with axes of length $\sqrt{\lambda_i}$.

5.2.3 Hand Shape

As stated in Section 2.1, there are only few basic hand shapes. A simple approach to create an appearance-based feature for hand shapes is to use distances to clusters of hand patches.

The clustering is done before the training of the visual models. The tracked hand patches of all frames in the training set are used to find clusters of hand shapes. In the presented system, the LBG-clustering algorithm [Linde & Buzo$^+$ 80] is used. This algorithm calculates image clusters by iteratively splitting Gaussian densities, starting with one density that is calculated using all images. Examples of estimated hand shape clusters are shown in Figure 5.5. PCA can be applied to the input images before clustering to reduce the dimensionality of the clusters.

The hand shape feature for a hand patch is composed of the Mahalanobis distances

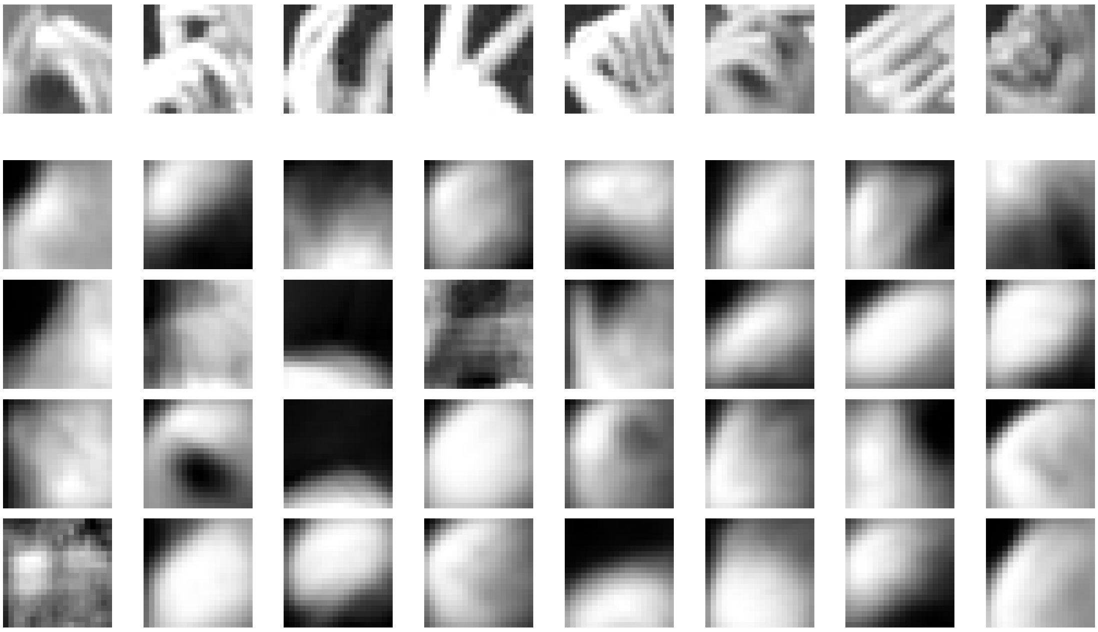


Figure 5.5. First row: examples of input images. Other rows: cluster centers of hand patches

between this patch and all clusters.

Differences in hand size are a problem for this approach. The hand size differers not only between different signers but also because of changing distances between hand and camera.

5.3 Non-Manual Features

The analysis of facial expressions, the detection of eye gaze and lips, and other non-manual features is complex, while the advantage of these features for sign language recognition is reported to be relatively low [Canzler 05]. Therefore, this work investigates only a simple appearance-based feature for non-manual components.

Mean face difference images (MFDI) are difference images between the mean face and the tracked face patch. The mean is computed for the face patches of a complete sequence, i.e. the frames of a sentence. Examples for mean face difference images are shown in Figure 5.6. These difference images show deviations of the face to the "normal" face of the signer. The dimensionality of this face feature can be reduced by applying PCA to the difference images.

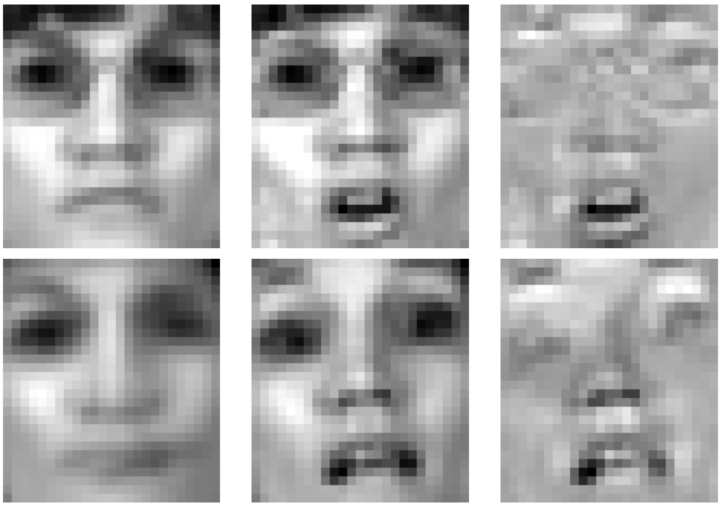


Figure 5.6. Examples of two mean face difference images. From left to right: Mean face, face patch, mean face difference image

5.4 Feature combination

The features presented in this chapter can be concatenated to composite feature vectors in order to model the different aspects of signs.

Another type of feature combination is often applied in speech recognition. Succeeding feature vectors are combined and LDA is applied to find an optimal combination of these features [Haeb-Umbach & Ney 92]. Feature vectors in a sliding window are combined to one feature vector by concatenating the individual vectors:

$$\begin{pmatrix} \cdots \\ x_{t-1} \\ x_t \\ x_{t+1} \\ \cdots \end{pmatrix}$$

The concatenated vector is then transformed with LDA.

A critical point is the size of the sliding window, because for an increasing window size an increasing amount of training data is needed [Katz & Meier$^+$ 02].

As mentioned in Section 5.1.2, LDA requires class information and depends on the amount of training data. Therefore, the application of PCA instead of LDA for feature combination has been investigated in this work as well.

Another possibility to use several features is the combination of visual models, like the combination of acoustic models in speech recognition [Zolnay & Schlüter$^+$ 05]. Therefore, visual models are trained independent of each other with different types

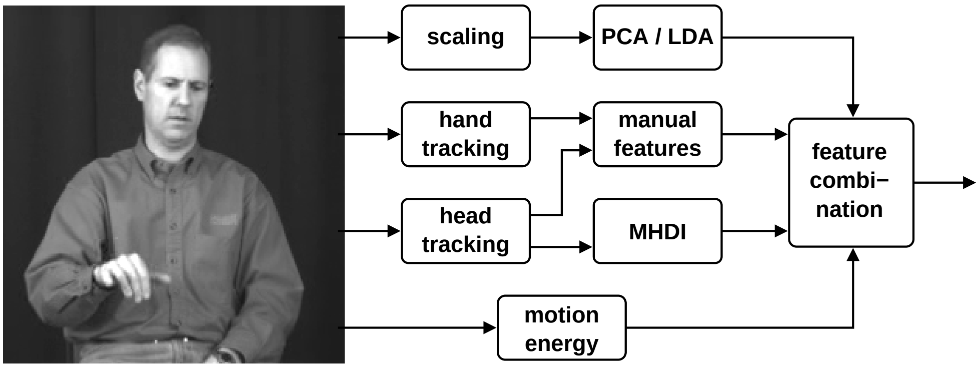


Figure 5.7. Overview on the feature extraction process

of features. These visual models are then combined for the recognition, by replacing the emission probabilities of the visual models. For a HMM state s and M models $p_m(x|s)$, the emission probability is:

$$p(x|s) = \prod_{m=1}^{M} p_m(x_m, s)^{\lambda_m} \ , \tag{5.5}$$

where λ_m is a model weight, and x_m is model specific feature vector. The feature vector x is here composed of different types of feature vectors $x = (x_1, \ldots, x_m, \ldots, x_M)$.

Figure 5.7 depicts the overall feature-extraction process. Features are extracted in several steps by different modules. Input frames of the videos are processed by two tracking algorithms, one to track the signer's dominant hand (Section 4.3) and the other one to track the head (Section 4.4). The extracted regions and positions are then used to compute manual features (Section 5.2) and non-manual features (Section 5.3). A scaled version of the input frames is transformed by either LDA or PCA to a feature vector of lower dimension (Section 5.1). Additionally, the motion energy in consecutive frames is measured (Section 5.1.3). All features are then concatenated to a composite feature vector. This feature vector can be combined with surrounding feature vectors by concatenation and application of PCA or LDA (Section 5.4).

Chapter 6

Databases

Databases for the evaluation of vision-based sign language recognition systems consist of videos showing a person that performs single signs or sign language sentences. Each video is annotated with data describing the performed signs. As mentioned in Section 3.6, all research groups use different databases for the evaluation of their systems. These databases differ in many aspects, for example in language, in vocabulary and in the number of signers. No database containing continuous sign language that has been used by other research groups for automatic sign language recognition is publicly available, except for the I6-BOSTON201 database mentioned in Section 6.1.

The RWTH-Boston-104 database is used for the evaluation of the methods presented in this work. This database is described in Section 6.1. The database used for the evaluation of the tracking methods is described in Section 6.2.

6.1 RWTH-Boston-104 Database

The RWTH-Boston-104 database is based on a sign language database published by the National Center for Sign Language and Gesture Recognition of the Boston University[1]. It has been recorded mainly for research on the syntactic structure of ASL [Neidle & Kegl$^+$ 99]. Thus, the data is not optimized for recognition tasks and the data is more realistic than databases recorded for the purpose of sign language recognition, e.g. the database used in [Bauer & Hienz$^+$ 00]. The original annotation provided by the Boston University has been created using SignStream$^{\text{TM}}$ and includes many aspects of sign language [Neidle 01]. At the Chair of Computer Science 6 of the RWTH Aachen University, 201 sentences have been composed for a new database. The annotation has been revised to fulfill the requirements of a sign language recognition system.

An earlier version of this database, called "I6-BOSTON201" was presented in [Zahedi & Dreuw$^+$ 06] and is publicly available[2].

The RWTH-Boston-104 database consists of 201 annotated videos of ASL sentences. The sentences have been performed by three signers (two women, one man). The

[1] http://www.bu.edu/asllrp/ncslgr.html

[2] http://www-i6.informatik.rwth-aachen.de/~zahedi/databaseBOSTON201.html

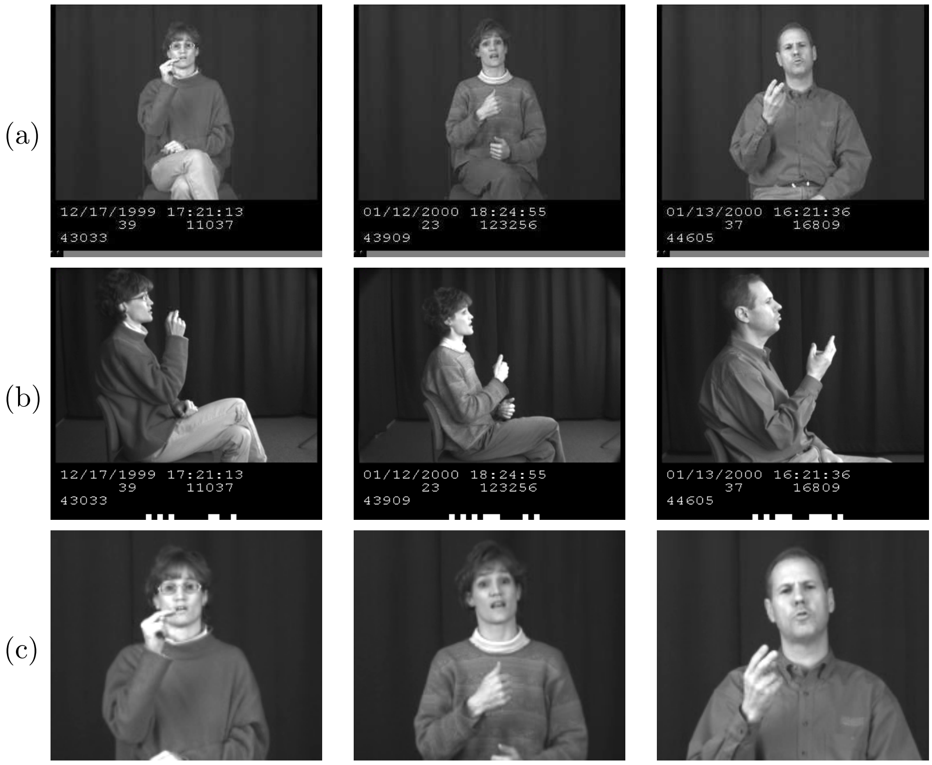


Figure 6.1. Sample frames of the RWTH-Boston-104 database: (a) frontal view, (b) side view, (c) extracted part of the frontal view

videos have been captured simultaneously with four stationary standard cameras. Two cameras forming a stereo pair show the frontal view of the signers. Another camera is located at the side of the signers. The fourth camera captures only the face of the signer. The videos are captured at a frame rate of 30 frames per second and at a resolution of 312 × 242 pixels. All cameras, except the face camera, have captured gray-scale videos. Sample frames are shown in Figure 6.1.

Most of the experiments presented in Chapter 7 have been carried out using one of the frontal cameras. The upper center part of the frames (size: 195 × 165 pixels) is extracted (see Figure 6.1), because the lower part of the frames contains textual information about the frames and the left and right borders of the frames are unused. Additional experiments have been carried out using the videos of the side view (250 × 240 pixels).

The set of 201 sentences is divided into a training set and a test set consisting of 161 and 40 sentences respectively. The sentences share a vocabulary of 104 words in total, 103 of them occur in the training set. The word that occurs only in the test set,

Table 6.1. Corpus statistics for the training and the test set. OOV means out of vocabulary

	sentences		glosses			
corpus	total	unique	total	unique	singletons	OOV signs
training	161	121	710	103	27	-
test	40	35	178	65	9	1

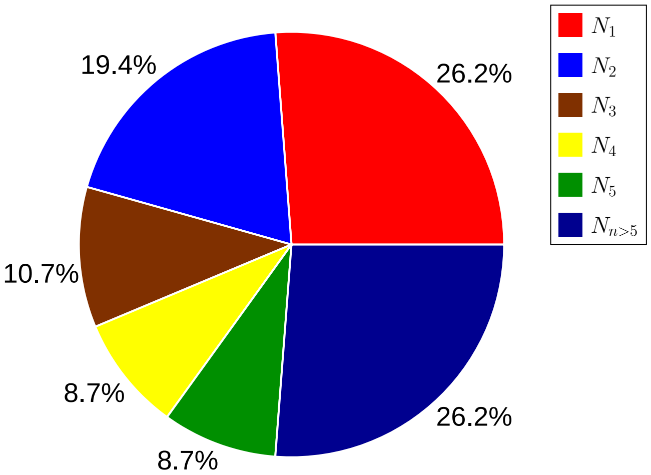


Figure 6.2. Word counts in the training set. $N_n := |\{w : N_w = n\}|$ denotes the number of words which occur n times. N_w is the number of occurrences of word w.

called out of vocabulary (OOV) word, cannot be recognized correctly, because there is no trained visual model for it. Detailed statistics about the training and the test set are shown in Table 6.1. The term *singleton* denotes words that occur only once in training. Nine of these singletons occur also in the test set.

The database is annotated with glosses (see Section 2.4). Start and end time of each sign in the training set have been detected manually and are included in the annotation data. Different pronunciations of words, i.e. different signs with the same meaning are annotated as well.

A disadvantage of this database is the high number of singletons. If a visual model is trained with only few observations, especially with only one observation, it is unlikely that this model matches unseen utterances. As can be seen in Figure 6.2, about 45 % of the signs in the training set occur only once or twice. The low number of observations of these signs also affects the other visual models, because continuous signing is used for the training of the visual models and the correct estimation of sign boundaries depends on the quality of all visual models.

Each signer occurs both in the training set and in the test set. The distribution of

Table 6.2. Number of sentences and words performed by each signer

	training set		test set	
signer	sentences	words	sentences	words
A	53	224	9	35
B	48	232	12	58
C	60	254	19	85

Table 6.3. Language model statistics for the RWTH-Boston-104 database. A zerogram language model is an uniform distribution of a-priori word probabilities. Sentence boundaries are modeled in the language model, too.

language model type	perplexity
zerogram	106.0
unigram	36.8
bigram	6.7
trigram	4.7

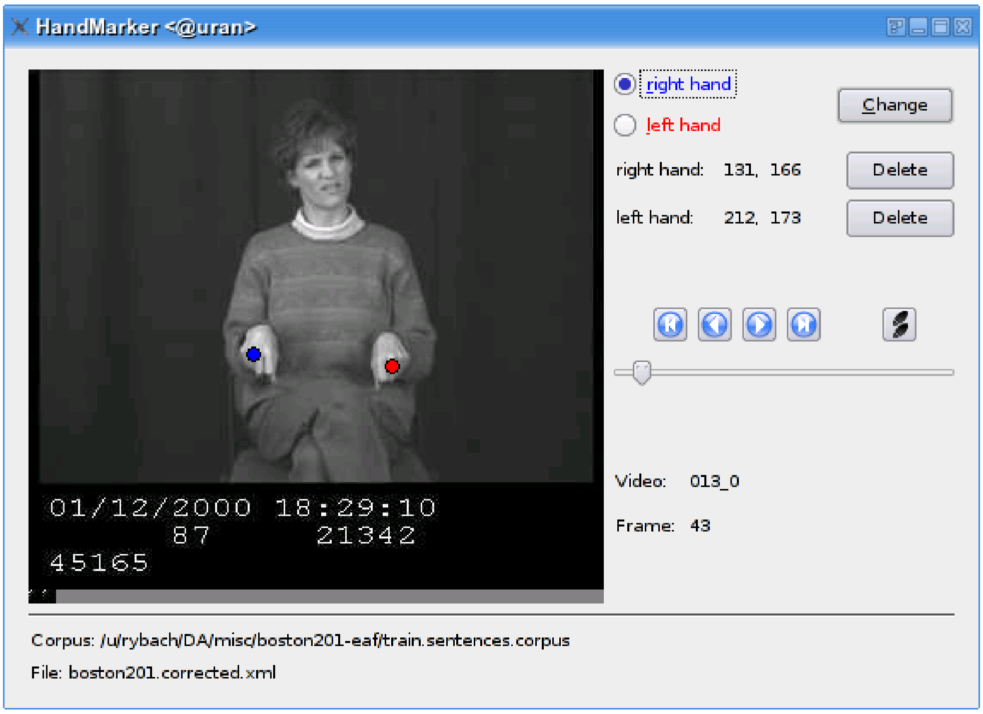
signers in the corpus is shown in Table 6.2.

The perplexity of the test corpus and different language models which have been estimated on the training corpus of this database are shown in Table 6.3. The perplexity of the bigram and trigram language model is low, because the sentences have a simple structure and 31 of the 40 test sentences occur in the training set. 60 % of all sentences start with the word “JOHN”, showing the simple structure of the sentences.

6.2 RWTH-Boston-Hands Database

There are many databases for the evaluation of tracking systems, e.g. the widely used PETS04 database [Fisher 04]. These databases are collected for tasks like human activity surveillance. The tracking algorithm presented in this work is specialized for tracking in sign language videos. It is not designed to handle leaving objects, multiple objects and other events that occur in person tracking tasks.

Therefore, a database for the evaluation of hand tracking methods in sign language recognition systems has been prepared. The RWTH-Boston-Hands database consists of a subset of the RWTH-Boston-104 videos. The positions of both hands have been annotated manually in 15 videos with an application developed for this work (see Figure 6.3). 1119 frames in total are annotated. All three signers of the RWTH-Boston-104 database appear in the annotated videos. Examples of annotated frames are shown in Figure 6.4.



right hand: 131, 166

left hand: 212, 173

Video: 013_0

Frame: 43

Corpus: /u/rybach/DA/misc/boston201-eaf/train.sentences.corpus

File: boston201.corrected.xml

Figure 6.3. Screenshot of the annotation tool for hand positions.

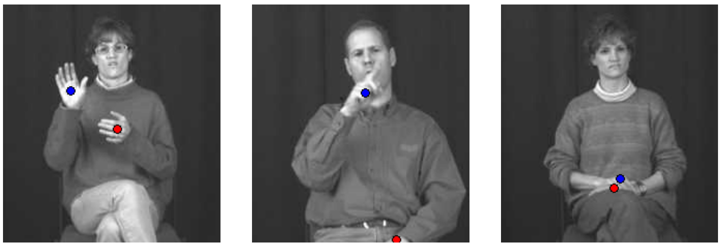


Figure 6.4. Sample frames of the RWTH-Boston-Hands database with annotated hand positions. Left and right hand are marked with red and blue circles respectively.

Chapter 7

Results

In this chapter the experimental results for the tracking methods and for sign language recognition are presented. The sign language recognition system, including the tracking algorithm, has been implemented in the framework of the RWTH large vocabulary speech recognition system [Kanthak & Molau$^+$ 00].

7.1 Tracking

The methods for hand tracking described in Section 4.3 are evaluated using the RWTH-Boston-Hands database (see Section 6.2). The results of the head tracking are presented only visually, because no database is available for the evaluation of head tracking methods in the context of sign language.

7.1.1 Performance Measurement

To be able to compare the different tracking methods and their parameters, two performance measurements are introduced. The *tracking error rate (TER)* measures the number of correctly positioned tracking windows. Only the center of the tracking window is used for the evaluation, because the annotations contain no information about the object size. The annotated hand position and the center of the tracking window do not need to be exactly identical, because the manual annotation is not exact and it suffices that the tracking window covers the object to be tracked. However, precise tracking is appreciated. Therefore, a tolerance τ is introduced defining the maximally allowed distance between annotated and tracked position (see Figure 7.1 for examples of this tolerance). For an image sequence X_1^T and corresponding annotated hand positions u_1^T, the TER of tracked positions $\hat{u}_1^T$ is defined as the relative number of frames where the Euclidean distance between the tracked and the annotated position is larger than or equal to τ:

$$TER = \frac{1}{T}\sum_{t=1}^{T} \delta_\tau(u_t, \hat{u}_t) \qquad \text{with} \qquad \delta_\tau(u,v) := \begin{cases} 0 & \|u-v\| < \tau \\ 1 & \text{otherwise} \end{cases} \tag{7.1}$$

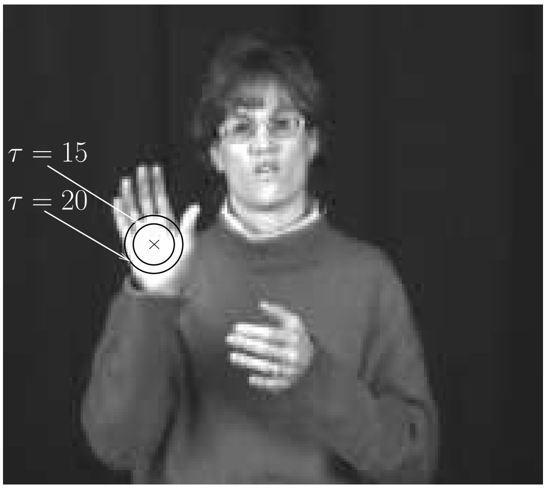


Figure 7.1. Tolerance for tracking results. The circles show the range of allowed distances between annotated and tracked position.

The other used performance metric is the average distance between annotated and tracked positions:

$$\mu_d = \frac{1}{T} \sum_{t=1}^{T} \|u_t - \hat{u}_t\| \tag{7.2}$$

These measurements do not distinguish between false positive and false negative detections, as it is done in other works concerning tracking methods, because we assume that the tracked object does not disappear. The tracking algorithm does not account for leaving objects.

For the calculation of tracking error rate and average tracking distance, frames in which the hand is not visible are disregarded.

7.1.2 Hand Tracking

Experiments for different score functions (see Section 4.3) have been carried out using the same basic setup. All parameters are kept constant and only the score function is changed:

- The set of allowed predecessor positions $\mathcal{M}(u)$ is set to

$$\mathcal{M}(i, j) = \left\{(i + i', j + j') : -J \leq i', j' \leq J\right\} \qquad \text{with} \quad J = 10 \ , \tag{7.3}$$

- a window size of 20 × 20 pixels is used,
- the jump penalty weight $\alpha_{\mathcal{T}}$ is set to 0.1.

Table 7.1. Results for different score functions. All experiments use the same setup. The tracking error rate is given for different tolerances τ.

	TER [%]		
score function	$\tau = 15$	$\tau = 20$	μ_d
motion (4.16)	17.80	6.07	10.06
absolute motion (4.17)	14.97	6.28	10.38
squared motion (4.18)	14.14	3.87	9.17
total motion (4.19)	11.83	3.56	8.72
constant background (4.21)	10.05	2.30	7.92

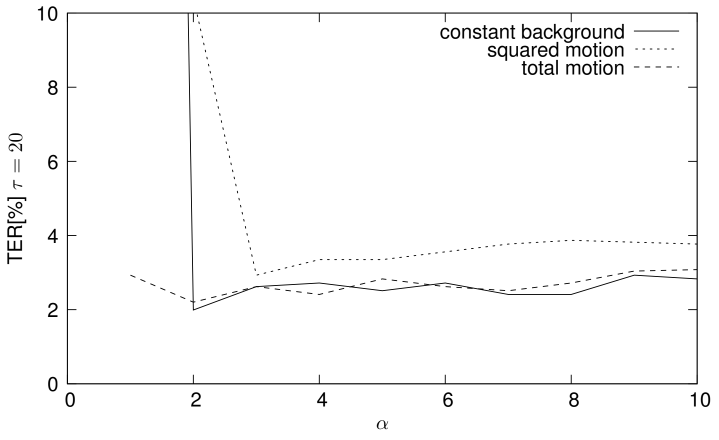


Figure 7.2. Results for the combination of score function (4.20) with other score functions. α is the weighting factor of the respectively other score function.

Table 7.1 shows the results of these experiments. The score function (4.20) is not included in this series of experiments with single score functions, because it needs an initialization with another score function. The score functions (4.16) and (4.17) use a linear relationship between image difference and score, while the other score functions use squared differences. The score functions that use squared differences perform better, as they penalize large differences more than small differences.

To avoid an initialization of the tracking, the "constant object" score function (4.20) can be used in combination with other score functions. Experiments with the combination of score function (4.20) and the best three scoring functions of Table 7.1 were carried out, using different weights. The results of these experiments are shown in Figure 7.2, the detailed results for the best weight of each combination is shown in Table 7.2.

Table 7.2. Results for the combination of score function (4.20) with other score functions. α is optimized (see Figure 7.2).

		TER [%]		
score function	α	$\tau = 15$	$\tau = 20$	μ_d
squared motion (4.18)	3	12.57	2.93	8.54
total motion (4.19)	2	9.74	2.20	8.27
constant background (4.21)	2	8.80	1.99	7.67

A weight of $\alpha = 0$ in Figure 7.2 does not produce meaningful results. If both score functions are weighted equally ($\alpha = 1$), only the combination with the total motion score function (4.19) yields a tracking error rate in a reasonable range. In the other combinations the constant object score functions seems to dominate the score. The combination of the constant background score function and the constant object score function (4.24) yields the best results as it considers both object and background constraints. A weighting of 1 : 2 for (4.20) and (4.21) respectively produces the fewest errors. The usage of higher weights for the constant background score function converges to the error rate of the experiment using only this score function. Other parameters of the tracking algorithm are evaluated using this combination of score functions.

Figure 7.3 analyzes the influence of the jump penalty weight. It can be seen that jump penalties larger than 0.3 deteriorate the results. The benefit of the jump penalty is obvious as it decreases the error rate from 7.4 % with $\alpha_{\mathcal{T}} = 0$ to 1.6 % with $\alpha_{\mathcal{T}} = 0.3$. Without a jump penalty the path of object positions is not required to be smooth, i.e. the object is allowed to "jump" inside the area of allowed predecessor positions.

The tracking window size is varied in Figure 7.4. It can be seen that a window size between 20×20 and 25×25 is a good estimate. This window size corresponds to the average hand size, as can be seen in Figure 7.1.

The maximum jump width J determines the set $\mathcal{M}(u)$ of allowed predecessors (see Equation 7.3). Figure 7.5 shows that a maximum jump width between 10 and 11 is optimal with respect to the tracking error depending on the tolerance. The maximum jump width should match the maximal hand speed divided by the frame rate of the recording. The speed of the tracked object is usually not known in advance, thus a high maximum jump width can be chosen, because it has mainly impact on the computation time and not on the error rate.

Replacing the Euclidean distance in the calculation of the difference between two tracked windows by the tangent distance increases the error rate, as shown in Table 7.3. These additional errors may result from the higher tolerance of the tangent distance regarding image transformations, especially translation.

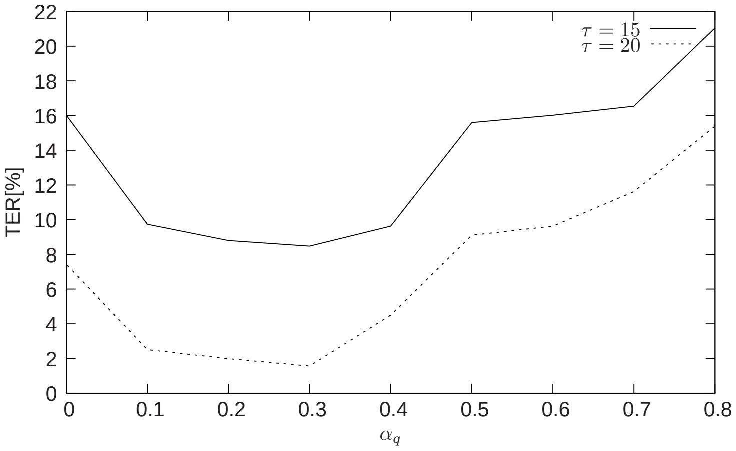


Figure 7.3. Results for different jump penalty weights α_q. (4.24) is used as the score function.

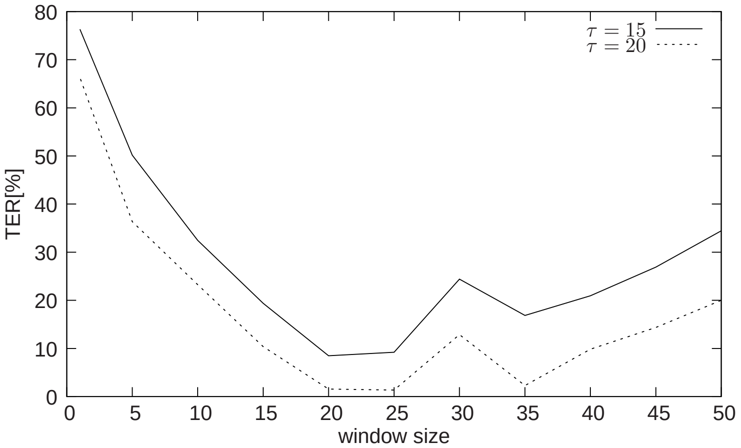


Figure 7.4. Results for different tracking window sizes. (4.24) is used as score function with a jump penalty weight of 0.3

Table 7.3. Results for different distances in (4.20). The score weights α are optimized for both distances separately. Score function (4.24) is used with a window size of 20×20 and a jump penalty weight of 0.3.

		TER [%]		
distance	α	$\tau = 15$	$\tau = 20$	μ_d
Euclidean distance	2	8.48	1.57	7.52
Tangent distance	6	9.01	3.25	7.93

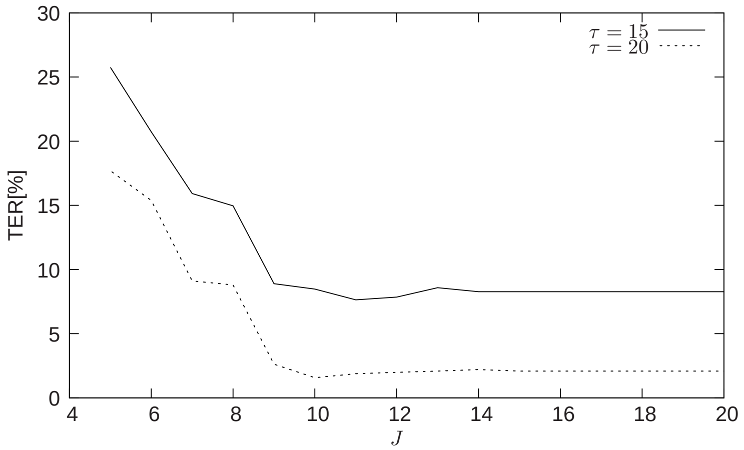


Figure 7.5. Results for different sets of allowed predecessors. J is the maximum jump width. (4.24) is used as score function with a jump penalty weight of 0.3

Table 7.4. Comparison of local decisions to decisions on the whole sequence. Score function (4.19) is used.

	TER [%]		
decision based on	$\tau = 15$	$\tau = 20$	μ_d
complete sequence	11.83	3.56	8.72
single frames	33.82	23.66	23.67

The dependency of the error rate on the tolerance τ is shown in Figure 7.6. For tolerances larger than 20 the error rate is very low. The usage of tolerances lower than 10 does not give realistic results, because in the presented system the tracking window is required only to cover the object to be tracked and not to be centered on an exact point.

Table 7.4 shows that the approach of avoiding local decisions works well. If the best object position is determined for each frame separately, the error rate increases significantly.

The following parameters have been proven to yield good results in the experiments presented in this section. Thus, they are used in the experiments for sign language recognition:

- score function “constant window and constant object” (4.24) with a weighting of $\alpha = 2$ for the “constant window” part.

- window size 20×20

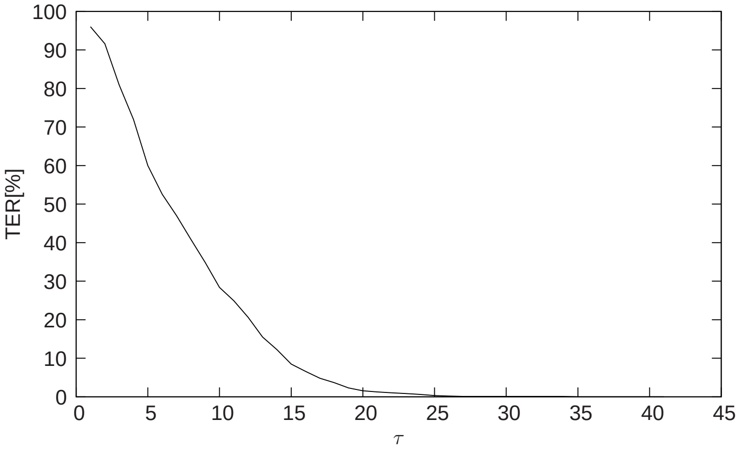


Figure 7.6. Results for different tolerances τ. The best settings for score function (Equation 4.24) described in Section 7.1 are used.

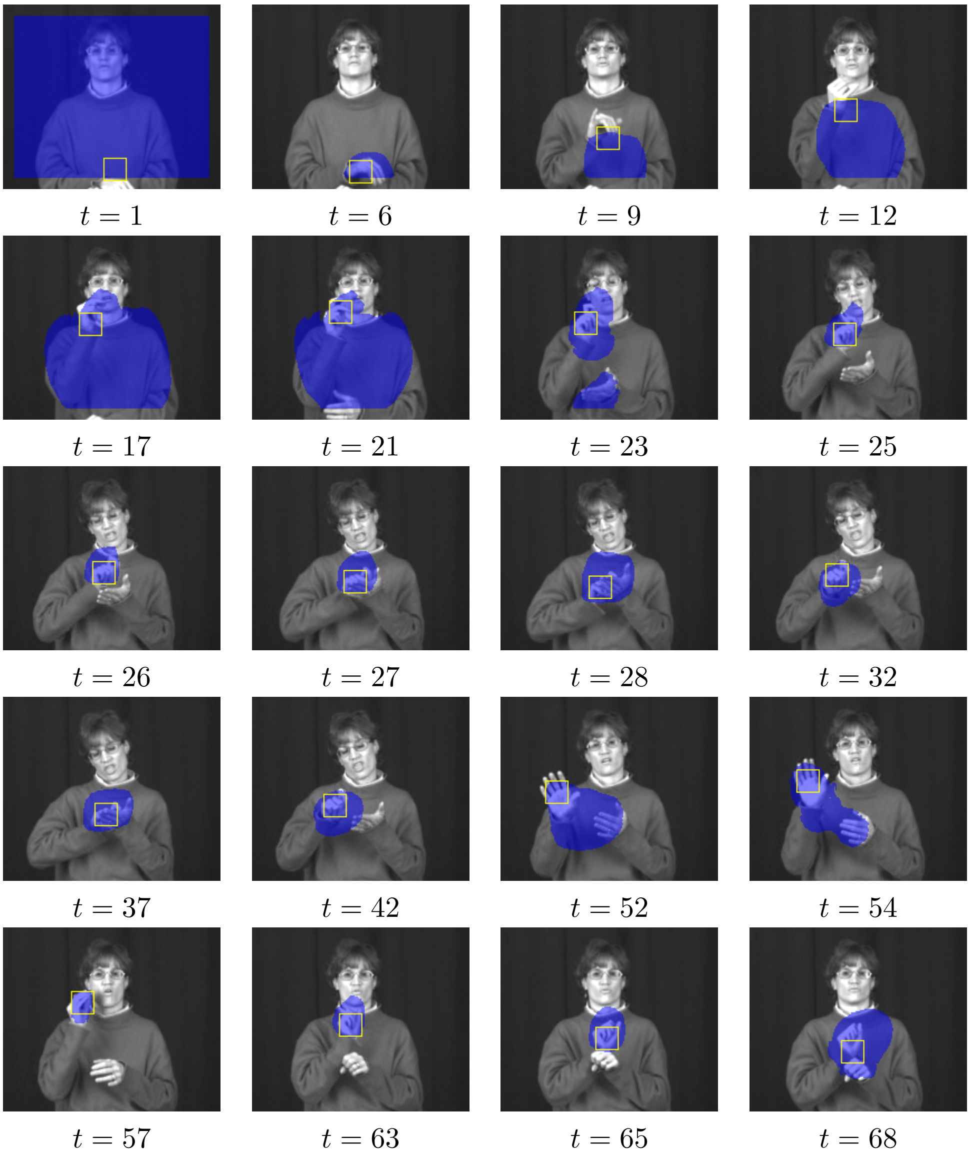
- jump penalty weight $\alpha_{\mathcal{T}} = 0.3$
- maximum jump width $J = 10$

Sample frames of an experiment with these settings and activated pruning can be found in Figure 7.7. It can be seen that the set of considered hand positions becomes small after a few frames ($t = 6$). If the hands are crossing ($t = 26 \ldots 42$), the set of considered object positions includes the non-dominant hand, but the tracked window stays on the dominant hand. In the first few frames the hand moves very fast ($t = 9 \ldots 12$), so that the maximum jump width is not large enough to allow the tracking window to cover the hand. However, after a few frames the tracking window catches up with the hand.

7.1.3 Head Tracking

The results of the head tracking experiments are evaluated only visually. As the RWTH-Boston-104 database consists of gray level videos, the skin color score function cannot be used. Instead, a brightness score is used, which is simply the sum of all pixel values in the tracking window. Results of experiments using this score (see Figure 7.9) show that the tracking window is not centered on the face. Instead, more homogeneous regions, like the neck, or brighter regions, e.g. the white collar of one of the signers, are detected. In Figure 7.8 it can be seen that the tracking becomes wrong when the face is partially occluding with a hand.

The eigenface scoring function alone does not always detect the face, depending on facial expressions and the signer's cloth (see Figure 7.9). The combination of these two



$t = 1$ $t = 6$ $t = 9$ $t = 12$

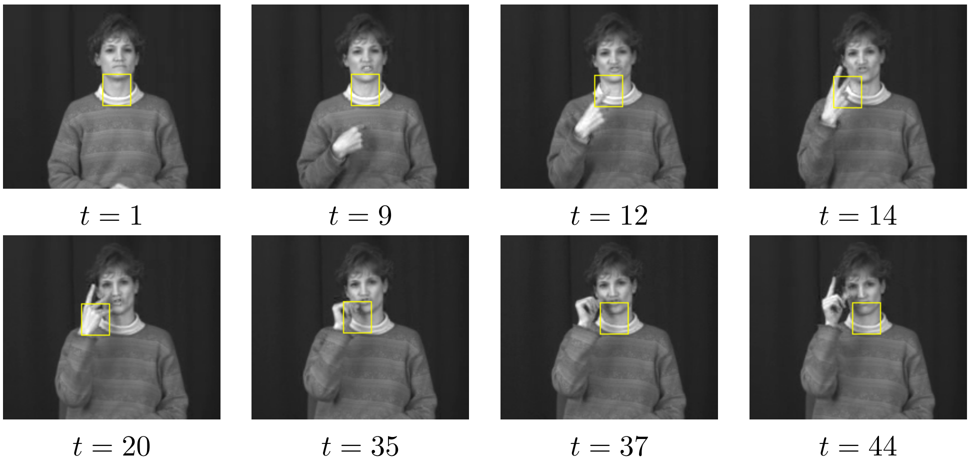
$t = 17$ $t = 21$ $t = 23$ $t = 25$

$t = 26$ $t = 27$ $t = 28$ $t = 32$

$t = 37$ $t = 42$ $t = 52$ $t = 54$

$t = 57$ $t = 63$ $t = 65$ $t = 68$

Figure 7.7. Sample frames of a hand tracking experiment with activated pruning. The tracking window is marked with a yellow rectangle. The active hypotheses, i.e. positions that are considered as possible object positions in each frame, are marked as a blue region.



$t = 1$ $t = 9$ $t = 12$ $t = 14$

$t = 20$ $t = 35$ $t = 37$ $t = 44$

Figure 7.8. Sample frames of a head tracking experiment using a brightness scores only. The tracked window is marked with a yellow rectangle.

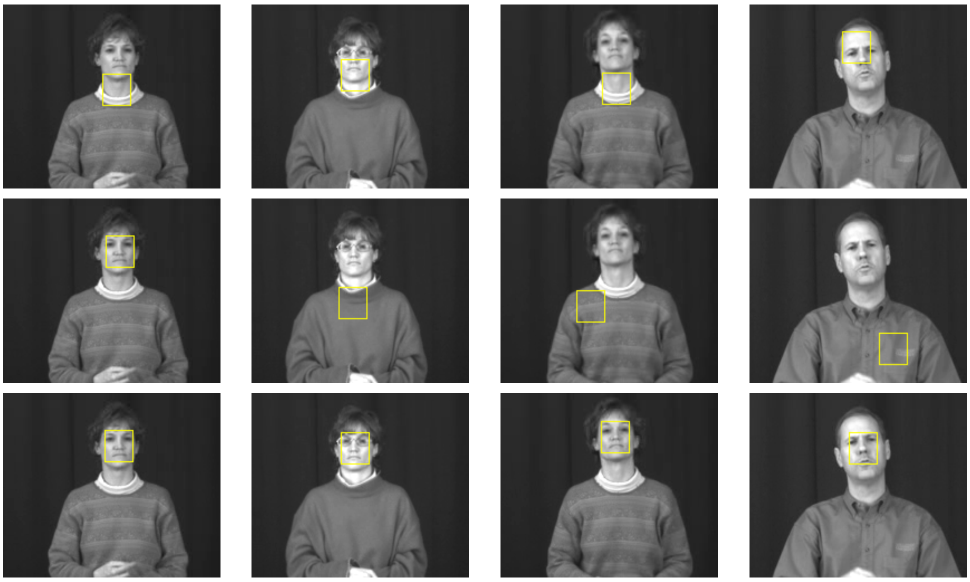
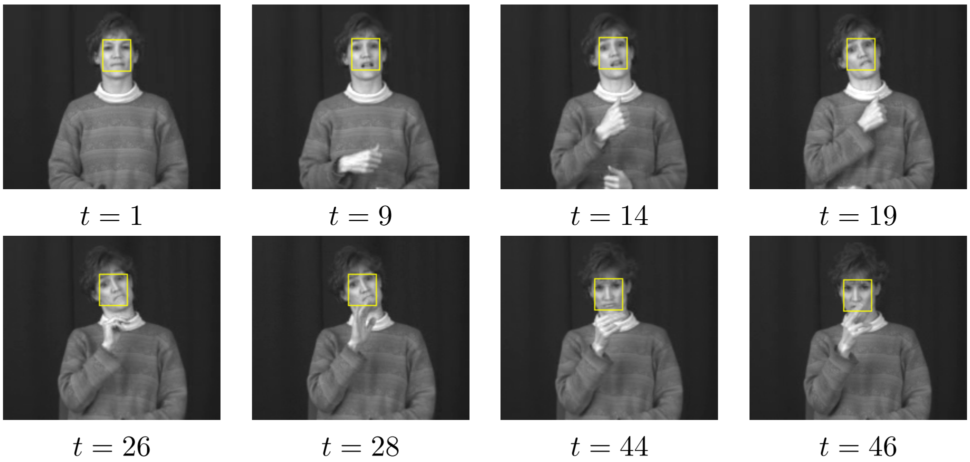


Figure 7.9. Sample frames of experiments with different score functions for head tracking in different videos. The following score functions are used (from top to bottom): brightness, eigenface, combination of brightness and eigenface.



$t = 1$ $t = 9$ $t = 14$ $t = 19$

$t = 26$ $t = 28$ $t = 44$ $t = 46$

Figure 7.10. Sample frames of a head tracking experiment using a combination of the eigenface scoring function with a brightness score. The tracked window is marked with a yellow rectangle.

score functions compensates the errors of both. Facial expressions and head rotations are tolerated as shown in Figure 7.10.

7.2 Sign Language Recognition

In this section, the results of the sign language recognition experiments are presented. All experiments have been carried out using the RWTH-Boston-104 database. The system is trained to recognize glosses without explicit modeling of inflections and incorporations. Special properties of sign language like indexing are not considered. Thus, pointing gestures are recognized as a pointing gloss without information about subject or direction of the sign.

First, parameters for word modeling are investigated. These parameters are then used in the experiments comparing different features and the combination of features.

7.2.1 Word Modeling

The duration of signs differs significantly. For example, the sign "BUY" has an average length of 0.18 seconds (6 frames), while an utterance of the sign "LEG" can take up to 0.96 seconds (29 frames). Small differences between the length of utterances are compensated by the HMM, but different lengths of the signs have to be modeled by the number of HMM states.

The relationship between average sign length and number of states is determined empirically. The number of states can be set to the same value for all signs or indi-

Table 7.5. Results for different numbers of HMM states for the visual models. The average length per state is the average sign length divided by the number of states of its visual model.

avg. length per state [ms]	avg. frames per state	WER [%]
30	1.0	43.3
40	1.3	37.1
50	1.7	39.3
60	2.0	38.8
90	3.0	51.1
fixed: 3 states / word	6.6	44.9
fixed: 6 states / word	3.3	38.7

vidually for each sign. The individual number of states of each word is adapted to the average length the word in the training set. The average length of a sign is calculated from the start and end times included in the annotation data of the database.

A basic setup is chosen for the experiments described in the following. The frame is scaled to a size of 32×32 pixels and transformed with LDA to 90 features. A trigram language model is used.

Table 7.5 shows that an average length of 40 ms per state yields the best error rate. If all words have the same number of states, i.e. word length is not modelled, the error rate is higher. The average number of frames per state for the fixed length models is not directly comparable to the variable length models, as frames are distributed over all states of all words in the training corpus. Thus, long words have too few states and short words have too many states.

If the number of states for the visual models is too low, the models of the states tend to be too generic, because the features of many succeeding frames will be mapped on one state. On the other hand, if the model has too many states, it may happen that the number of frames of an observation is lower than the number of states, resulting in a wrong model or in an error of the training algorithm. The usage of skip transitions can avoid some problems with too long models, as described in Section 7.2.4.

Different signs can convey the same meaning, i.e. there are different pronunciations of a sign [Zahedi & Keysers$^+$ 05]. As mentioned in Section 6.1, these pronunciation variants are labeled in the database. The length of each pronunciation is estimated separately. The results in Table 7.6 show that the consideration of pronunciations improves the results, because pronunciations of a sign differ in visual appearance and in duration.

Table 7.6. Results with and without consideration of pronunciations.

words	pronunciations	WER [%]
103	103	39.9
103	112	37.1

7.2.2 Features

The simplest features are the appearance-based features based on complete frames. The results of experiments with the frame itself as feature and transformed representations of it are shown in Table 7.7. The frames are downscaled to 32 × 32 pixels. The table shows also the result of an experiment where two transformation are applied. PCA is used to reduce the dimensionality of the features, then the reduced features are transformed with LDA. The other way around, first LDA then PCA, is not reasonable, as the results show that the dimension reduction of PCA produces better results than LDA. Thus, a further dimension reduction with PCA of LDA transformed features is not expected to produce better results. The results of the feature dimension reductions of LDA and PCA are analyzed in Figure 7.11. Both dimension reduction techniques improve the results, because less parameters of the emission probabilities of the visual models have to be estimated with the low amount of training data. The application of PCA yields the best results. The difference between LDA and PCA can be seen in the visualization of both in Figure 7.12. The LDA transformation discards pixels in center of the image which have a low between class variance and keeps the border pixels. In contrast, the PCA discards these border pixels because of their low total variance. Most border pixels belong to the background and do not change in the videos. Thus, they are not likely to be useful for recognition. As mentioned in Section 5.1.2, the calculation of a robust LDA relies on correct class information and a large amount of training data, which is not given in the RWTH-Boston-104 database. The visualization shows also the so-called “salt-and-pepper” structure [Hastie & Tibshirani^{+} 03, page 398] of the LDA projection vectors. The combination of PCA and LDA does not improve the results.

Scaling the frames to 32 × 32 pixels introduces a slight distortion, because the original aspect ratio is not kept. Experiments with other scalings in Table 7.8 show that undistorted frames yield better results, but higher resolutions do not give further improvements. The usage of larger frames requires the estimation of a larger covariance matrix for the calculation of the PCA transformation. The more parameters have to be estimated the more training data is needed. If the PCA transformed frame is combined with other features, the benefit of the undistorted frames vanishes, as can be seen in Table 7.14 (page 66). Thus, a PCA transformation of the 32 × 32 frame to 110 features is used in further experiments, abbreviated as *PCA-frame*.

Table 7.7. Results using the original frame downscaled to 32×32 pixels and LDA, PCA transformations.

	dimensionality		
transformations	after PCA	after LDA	WER [%]
-	-	-	48.3
LDA	-	80	36.0
PCA	110	-	27.5
PCA, LDA	110	80	30.3

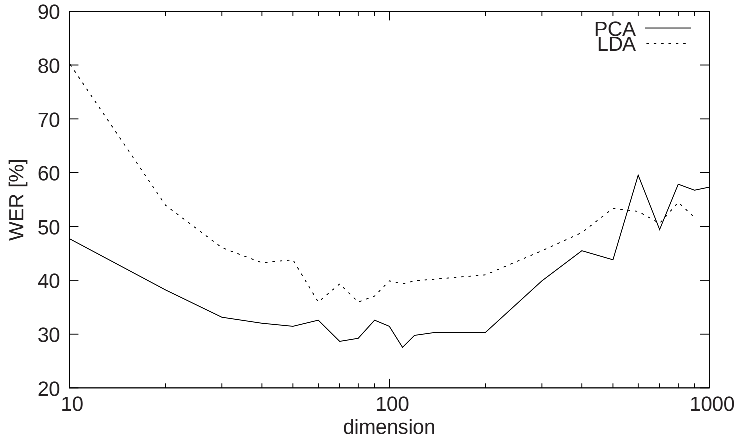


Figure 7.11. Results for different dimensionalities of PCA, LDA transformations.

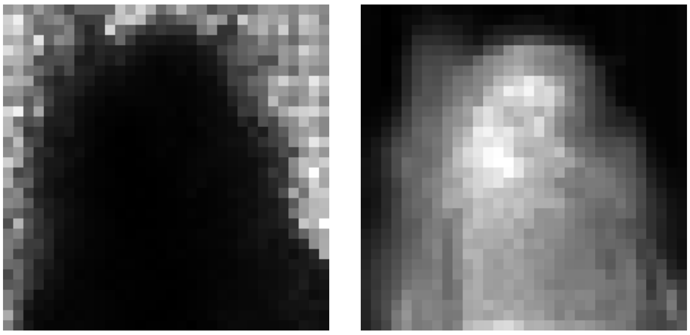


Figure 7.12. Visualization of a LDA (left) and PCA (right) transformations.

Table 7.8. Results of experiments with PCA transformed frames for different frame sizes. The dimensionality of the PCA transformed features is optimized for each frame size.

frame size	dimensionality after PCA	WER [%]
32×32	110	27.5
38×32	110	24.2
48×32	110	25.8
48×48	100	27.5
57×48	100	27.5
64×64	200	29.2

Table 7.9. Results for combinations of manual features. The error rates are given for successively added features. The results for all combinations can be found in Appendix A in Table A.1.

features	dim. of the feature vector	WER [%]
hand position u_t	2	59.6
+ hand motion m_t with $\Delta = 1$	4	48.9
+ hand motion m_t with $\Delta = 2$	6	46.1
+ motion energy e_t	7	42.1

Results of experiments with manual features, motion energy, and combinations are shown in Table 7.9. As one can expect, the usage of these simple features alone does not produce good results. However, it can be seen that the combination of hand position and hand motion features produces relatively good results, whereas position alone does not achieve good results. The best results are obtained when all four features are combined.

Manual features and motion energy can be combined with the PCA transformed frame (32×32 pixels, transformed to 110 features). The results of the experiments in Table 7.10 show that all manual features except of the combination position with motion slightly improve the result. The largest improvement is observed when both hand motion features are used. The hand position is encoded in the image, whereas information about movements cannot be observed in single frames. Furthermore, the hand position heavily depends on the speaker and his distance to the camera. Additional usage of motion energy does not yield further improvements.

Table 7.11 shows that the combination of PCA transformed frames and eigenvalue trajectory features outperforms the results of both PCA transformed frames and combinations with manual features (see Table 7.10). The trajectory features contain more

Table 7.10. Results for the combination of PCA-frames and hand position u_t, hand motion m_t, motion energy e_t. The results for all combinations can be found in Appendix A in Table A.2.

features	dim. of the feature vector	WER [%]
PCA-frame	110	27.5
PCA-frame + u_t	112	25.3
PCA-frame + m_t with $\Delta = 1$	112	27.5
PCA-frame + m_t with $\Delta \in \{1, 2\}$	114	24.2
+ e_t	115	24.7
+ u_t	117	27.0

Table 7.11. Results for combination of PCA-frames with trajectory eigenvalues λ_1, λ_2 and eigenvectors v_1, v_2 (sorted by decreasing eigenvalues). The window size is set to $2\Delta + 1 = 5$. Results for experiments with single trajectory features and for all combinations can be found in Appendix A in Table A.3.

features	dim. of the feature vector	WER [%]
PCA-frame	110	27.5
PCA-frame + λ_1, λ_2	112	23.6
PCA-frame + v_1	112	26.4
+ v_2	114	27.0
PCA-frame + λ_1, λ_2 + v_1	116	26.4

global information of the hand positions than the other manual features. The analysis of different sizes of the sliding window in Table 7.12 shows that a window size of 5 yields the best results for the eigenvalues feature. If the window is too large, the direction and the shape of the movement become imprecise.

In Table 7.13 it can be seen that the combination of PCA transformed frames, the trajectory eigenvalue feature, and manual features does not improve the results of the combination of transformed frame and trajectory eigenvalue feature. All combinations yield higher error rates than the best result in Table 7.12, two combinations yield even higher error rates than the usage of the transformed frame only.

The results of experiments with mean face difference images are shown in Table 7.15. Mean face difference images (MFDI) and PCA transformed MFDI alone do not yield good results, as no information about manual components is included. The combination of PCA transformed MFDI and PCA transformed frames yields an improvement in comparison to the error rate obtained with frame features alone. The best result

Table 7.12. Results for combinations of PCA-frames and trajectory features with different window size $2\Delta + 1$

	WER [%] obtained with		
Δ	λ_1,λ_2	v_1	λ_1, λ_2, v_1
1	26.4	25.3	26.4
2	23.6	26.4	26.4
3	25.3	27.0	24.7
4	24.2	28.1	26.4
5	28.7	27.0	25.8

Table 7.13. Results for the combination of PCA-frames, trajectory eigenvalue feature (window size 5), and manual features. All combinations include PCA-frames and trajectory eigenvalues. Results for all possible combinations can be found in Appendix A in Table A.4.

	dim. of the	
additional features	feature vector	WER [%]
-	112	23.6
u_t	114	28.1
m_t with $\Delta = 1$	114	27.0
m_t with $\Delta \in \{1, 2\}$	116	27.0
$+ e_t$	117	24.7
$+ u_t$	119	25.8

Table 7.14. Comparison of results obtained with combinations of manual features with PCA transformed frames of size 32×32 and 38×32.

	WER [%]	
features	32×32	38×32
PCA-frames	27.5	24.2
PCA-frames, m_t with $\Delta \in \{1, 2\}$	24.2	27.5
PCA-frames, trajectory eigenvalues	23.6	27.5

Table 7.15. Results for mean face difference images, PCA transformations of it (MFDI-PCA), and the combination with PCA-frames, trajectory eigenvalue features, and manual features. The dimensionality of the MFDI-PCA is given in brackets.

features		WER [%]
MFDI		56.2
MFDI-PCA (110)		54.0
PCA-frames, MFDI-PCA	(110)	35.4
	(55)	29.8
	(30)	26.4
	(20)	25.3
	(10)	28.1
	(5)	26.4
PCA-frames, MFDI-PCA	(20), trajectory	30.3
PCA-frames, MFDI-PCA	(20), m_t with $\Delta \in \{1, 2\}$	25.3

is obtained, if the number of MFDI features is much lower than the number of frame features resulting in a relatively low influence of the MFDI. Results obtained with the combination of MFDI and manual features or trajectory features are equal to or worse than the results obtained without additional features.

The usage of hand patch clusters for hand shape features deteriorates the results, as can be seen in Table 7.16. The hand patch clusters (see Figure 5.5, page 41) are visually not similar to real hand shapes, because the hand patches obtained with the used tracking method are not aligned to a fixed point of the hand and are not normalized in size.

In addition to the frontal view used so far, the RWTH-Boston-104 database provides also videos of the signers' side view. This perspective shows the vertical position of the hands, like the frontal view does, and the distance between hand and body which is not directly measurable in the frontal view. However, the hand position on the horizontal axis in front of the signer can not be quantified from this perspective with the used tracking method. The results in Table 7.17 show that the side view contains data which can improve the recognition performance. The usage of PCA transformed frames of the side view yields better results than the usage of the PCA transformed frontal view frames. A scaling of 38×32 improves the results even though it introduces a deformation. When trajectory features are added, results obtained with the eigenvalue feature are worse than those obtained with the trajectory of the frontal view, but the combination with eigenvalues and eigenvector yields better results than any combination of features of the frontal view.

The frames of the two cameras can be combined by either concatenating the PCA transformed frames or by applying one PCA transformation to feature vectors con-

Table 7.16. Results of experiments with hand patch clusters. The hand patches have a size of 20×20 pixels. Experiments have been carried out with clusters of untransformed hand patches (HP), clusters of PCA transformed hand patches (HP-PCA, 40 features), and in combination with PCA-frames.

features	number of clusters	WER [%]
HP	16	57.3
HP, PCA-frame	4	28.6
	8	31.5
	16	33.2
	32	34.8
	74	41.1
HP-PCA, PCA-frame	16	33.7

Table 7.17. Results obtained with features of frames captured by the side view camera. The extracted frame has a size of 250×240 pixels. Frame features are calculated from scaled versions of size 32×32, if not explicitly stated. Trajectory features are calculated with a window size of 5 frames unless otherwise noted.

features	WER [%]
PCA-frame	25.3
PCA-frame (frame size 38×32)	24.7
PCA-frame (frame size 48×48)	27.0
PCA-frame, trajectory eigenvalues	27.0
PCA-frame, trajectory eigenvector	25.8
PCA-frame, trajectory eigenvalues, eigenvector	23.0

sisting of concatenations of both frames. The results in Table 7.18 show that the concatenation of transformed frames produces more errors while the transformed concatenation yields an error rate slightly lower than the one obtained with the frontal view but higher than the error rate of using only the side view. However, the setup with two cameras is not likely to be found in other sign language databases. The three dimensional data obtained from a pair of stereo cameras is expected to provide similar or even more improvements, but was not investigated in this work.

The results presented in this section show that combinations of PCA-frames with hand motion features or trajectory eigenvalues yield good results, if the frontal view used. For the side view, the combination of PCA-frames with trajectory eigenvalues and eigenvectors performs best. These features will be considered in the following sections.

Table 7.18. Results obtained with combinations of both cameras. Combination is done by using either one PCA for the combination of frames or by concatenating PCA-transformed frames.

combination	WER [%]
concatenation of PCA-frames	34.8
one PCA transformation (dim.: 140)	27.0

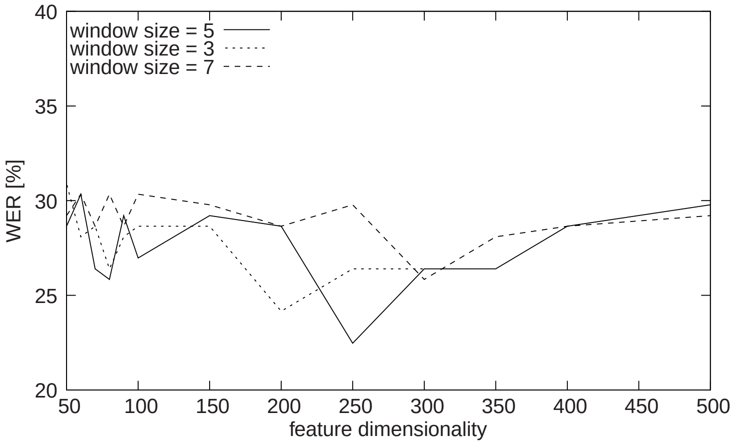


Figure 7.13. Results for feature combination of PCA transformed frames using LDA.

7.2.3 Feature Combination

Features of succeeding frames can be combined using both LDA and PCA (see Section 5.4). Parameters of these feature combinations are the number of frames combined and the dimensionality of the resulting feature vectors. Experiments have been carried out to compare both linear combination techniques and to find meaningful parameters. The comparison of the results in Figure 7.13 and Figure 7.14 shows that the application of PCA yields better results than the application of LDA. Both feature combination methods improve the results over those obtained with features of one frame. Furthermore, it can be seen that a window size of 5 frames gives the lowest error rates when using PCA. If too many frames are combined, coarticulation effects emerge, because frames belonging to adjacent signs influence the calculation of the features. The best result is obtained with a feature dimensionality of 100 using PCA, although each of the combined features has a dimensionality of 110.

Results of feature combinations for composite features are shown in Table 7.19. The feature combination improves all results obtained in experiments with the frontal view camera while the results obtained with the side view camera become worse. The best

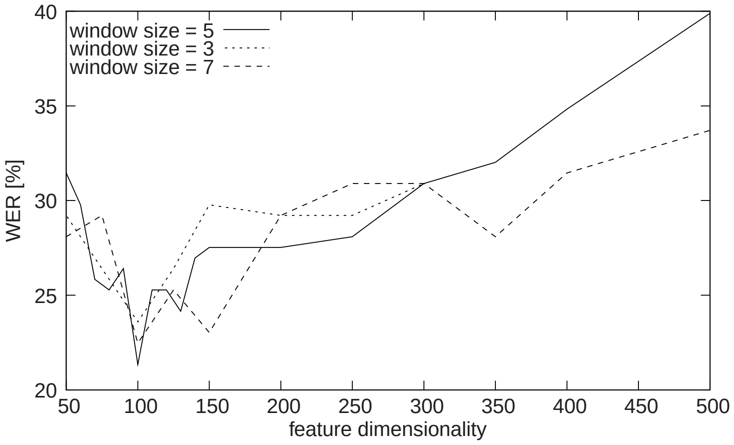


Figure 7.14. Results for feature combination of PCA transformed frames using PCA.

result in this series of experiments is obtained with the combination of side view frames and frontal view trajectory features.

7.2.4 HMM Topology

All experiments presented in the previous sections have been carried out without skip transitions in the HMMs, because they can produce errors in the training algorithm for the visual models. If skip transitions are allowed, HMM states can be skipped in the alignment of feature vectors to HMM states. If a state is skipped in the alignments of all training samples of a sign, the state is completely unseen in training and thus, the parameters of the emission probability of this state cannot be estimated. This situation is not unlikely, especially if only one training sample is available for a sign. To avoid this problem of unseen states while allowing skip transitions, one can set the transition probabilities of the skip transitions so low, that it is used in exceptional cases only.

The missing skip transitions can produce recognition errors, if the observed sign is shorter than the minimal length of the visual model. For example, a sign that occurs over a length of 4 frames cannot be recognized correctly with a 6 state visual model. Thus, allowing skip transition for recognition is expected to reduce recognition errors. This expectation is met, as can be seen in Table 7.20. The relatively high number of deletion errors compared to the number of insertion errors in the experiment without skip transitions indicates that the length of some recognized words is too long, such that surrounding words are not recognized. Deletion and insertion errors are more balanced when skip transitions are allowed. The results in Table 7.21 show that the usage of skip transitions improves also the recognition performance of setups with

Table 7.19. Results for feature combination of composite features using PCA. 5 succeeding feature vectors are combined and transformed to 100 features. Frames have a size of 32×32 unless otherwise noted.

camera	features	WER [%]
	PCA-frame	21.4
frontal	PCA-frame, m_t with $\Delta \in \{1, 2\}$	23.6
view	PCA-frame, trajectory eigenvalues	21.4
	PCA-frame, trajectory eigenvalues, eigenvectors	24.2
	PCA-frame	26.4
side	PCA-frame, m_t with $\Delta \in \{1, 2\}$	23.0
view	PCA-frame, trajectory eigenvalues	23.0
	PCA-frame, trajectory eigenvalues, eigenvectors	23.6
both	frontal view PCA-frame, side view trajectory eigenvalues	22.5
	side view PCA-frame, frontal view trajectory eigenvalues	20.2

Table 7.20. Comparison of results with and without skip transitions in training and recognition. The experiments use 5 combined PCA-frames (frontal view) as feature. Word errors are split in substitution errors (sub.), insertion errors (ins.), and deletion errors (del.).

skip transitions	word errors			
allowed in	del.	ins.	sub.	WER [%]
-	20	3	15	21.4
recognition	11	8	18	20.8
recognition, training	12	7	18	20.8

composite features and feature combination.

7.2.5 Language Modeling

The impact of the language model on the recognition performance can be analyzed by comparing results obtained with different types of language models. A zerogram language model assigns the same probability to all words, i.e. the structure of the language is not modeled. Unigram language models rate words by their frequency in the training set. More complex language models incorporate word sequences. Figure 7.15 shows that the usage of bigram and trigram language models improves the recognition performance significantly. The optimal language model scale is lower for zerogram and unigram language models, because they are less meaningful so the visual model must have more influence on the decision.

Table 7.21. Experiments with different features comparing results with and without skip transitions in training and recognition. All features include the PCA transformed frames. Feature combination is used in the two lower experiments.

	feature	WER [%]	
additional features	combination	w/o skips	skips
-		27.5	24.7
m_t with $\Delta \in \{1, 2\}$		24.2	22.5
trajectory eigenvalues	X	21.4	20.2
(side view) trajectory eigenval., -vec.	X	23.6	19.7

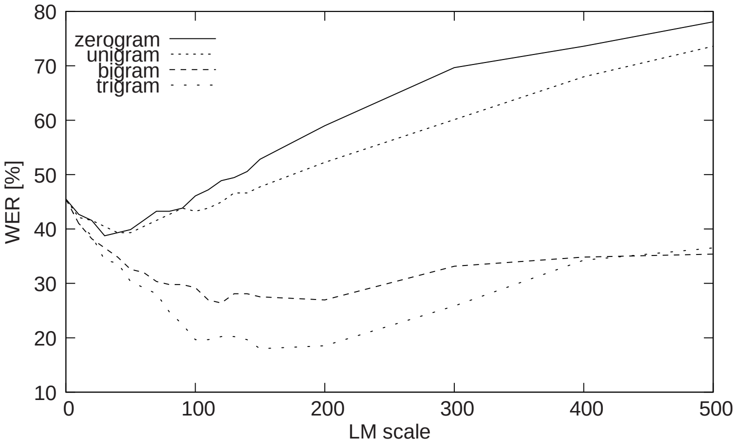


Figure 7.15. Comparison of the effect of different language models and different language model scales

One has to keep in mind that the vocabulary of 104 words is very small compared to those of current speech recognition systems and the structure of the sentences in the Boston-RWTH-database is rather simple. Thus, the obtained results are not directly transferable to corpora with larger vocabulary and more complex sentence structures. In speech recognition, where large corpora with a huge amount of data are used, the impact of the language model is higher than in the presented experiments.

7.2.6 Model Combination

As mentioned in Section 5.4, several visual models with different types of features can be combined. Models that yield good results when used alone are used for the combination (see Table 7.22). The results of the experiments with combined models are shown in Table 7.23. It can be seen that the combination of models can improve the results over the results obtained with single models.

Table 7.22. Description of the models that are combined in Table 7.23. Feature combination with a window size of 5 is used for all models.

model name	camera	features
model-1	frontal	PCA-frames, trajectory eigenvalues
model-2	frontal	PCA-frames, m_t with $\Delta \in \{1, 2\}$
model-3	side	PCA-frames, trajectory eigenvalues, eigenvector
model-4	side	PCA-frames, m_t with $\Delta \in \{1, 2\}$

Table 7.23. Results obtained with the combination of several visual models. The used models are described in Table 7.22. Skip transitions are used.

models	WER [%]
model-1, model-2	18.0
model-2, model-3	20.2
model-1, model-3	18.5
model-1, model-2, model-3	19.7

7.2.7 Cross Validation

All results presented in this chapter are obtained using the same partition of the available data into training and test sets. Due to the small amount of data it is not reasonable to split the data further to create an additional development set. Development sets are used to optimize the parameters of the recognition system, which is then evaluated using the optimized parameters on a separate evaluation set.

Due to the missing development set in the Boston-RWTH-database some of the parameters of the recognition system have been optimized on the test data. This practice may distort the results as the parameters are optimized for this specific test set and it is difficult to make a statement on the recognition performance in general.

In k-fold cross validation, the set of data is partitioned into k distinct subsets. Each of the subsets is used one time as test data while the other $k - 1$ sets have been used for training. Thus, k recognition experiments with different training and test sets can be carried out.

In this work, a 20-fold cross validation is used, resulting in test sets of 10 and training sets of 191 sentences. Details about the cross validation corpus can be found in Table 7.24. The number of singletons in the training corpus is lower than in the normal training corpus (see Table 6.1, page 47), because the training sets include more utterances. Visual models and trigram language models are estimated for each of the training sets. One half of the test sets was used to optimize two parameters, namely the language model scale and the word penalty, the optimized parameters are then

Table 7.24. Statistics about the cross validation corpus

	training sets			test sets		
	min	max	avg.	min	max	avg.
sentences	190	191	191	10	11	10
glosses	833	851	844	37	55	44
unique glosses	101	104	103	21	34	27
singletons	12	28	19	0	4	2
OOV glosses	-	-	-	0	3	1
perplexity	-	-	-	4.2	15.5	8.2

Table 7.25. Results of cross validation experiments using PCA-frames and trajectory eigenvalues (both from the frontal view camera) combined with a window size of 5. Skip transitions are allowed during recognition.

	dev. sets			eval. sets		
partitioning	min	max	avg.	min	max	avg.
1	21.6	57.5	39.9	27.5	61.5	43.5
2	25.0	61.5	43.2	21.6	55.1	37.3

used in the experiments on the remaining test sets.

The average error rates in Table 7.25 are significantly higher than those obtained using the normal training and test sets. It can be seen that the error rate depends on the used test set. The obtained results range between 22 % and 62 % error rate. The recognition experiments have been carried out using two different partitions of the test sets into development and evaluation set. In one partition, the first ten test sets are used for the parameter optimization, the remaining test sets are used for evaluation, and vice versa in the other partition. One half of the test sets obtains better results whether or not it is used for parameter optimization.

Several aspects are analyzed in order to find a reason for the huge differences between the results of the test sets. Figure 7.16 shows the number of sentences N_s that occur both in training and in test (sentences that consist of the same sequence of words, not the same utterance) as the result obtained with this combination. One can see, that there is no correlation between these two factors. If the word error rate would depend on N_s, the visual models would be sentence specific and not able to match words in an unseen context.

Another interesting aspect is the frequency of the test words in the training corpus, meaning how often a word has been seen in training. In Figure 7.17 the average frequency of all words and the average frequency of all unique words in a test set is

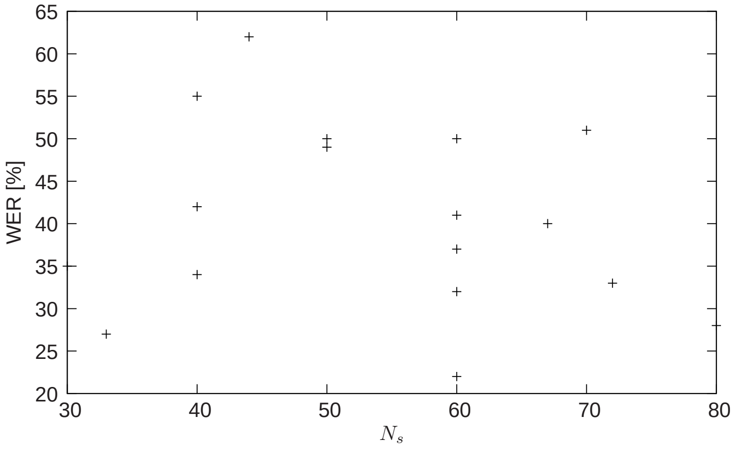


Figure 7.16. Results of the cross validation experiments. Each point represents a test set. On the x-axis the number of sentences in the test set that occur also in its training set (N_s) is listed and the error rate obtained with this test set on the y-axis.

Table 7.26. Summary of the results presented in this chapter.

features	result table	WER [%]
untransformed frames	7.7	48.3
PCA-frames	7.7	27.5
PCA-frames, hand motion	7.10	24.2
PCA-frames, trajectory	7.11	23.6
feature combination using PCA	7.19	21.4
skip transitions	7.21	20.2
model combination	7.23	18.0

plotted in combination with the obtained error rates. Also in this graph no correlation is noticeable. Thus, it is most likely that the different error rates are caused by the different compositions of words in the particular cross validation partitions.

7.2.8 Summary

The results presented in the different sections of this chapter are summarized in Table 7.26. In the progress of this work, the the presented features and methods are able to improve the error rate from 48 % to 18 %.

The best presented automatic sign language recognition system recognizes the sentences of the RWTH-Boston-104 test set with a word error rate of 18 % (see Table 7.23). The test set consists of 40 sentences, 22 of them are recognized correctly. In the re-

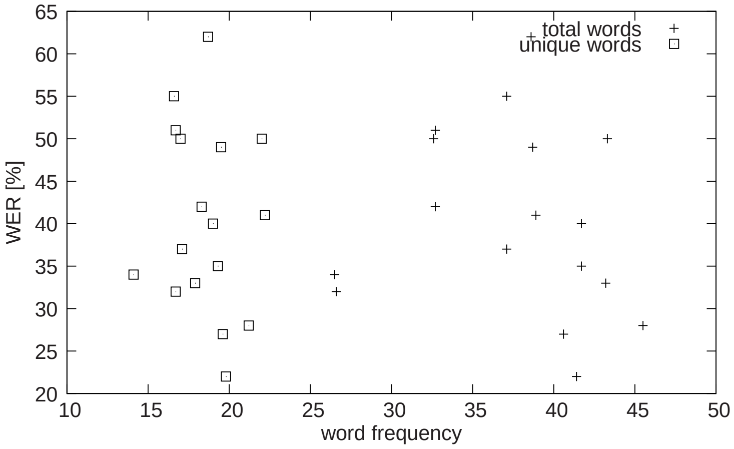


Figure 7.17. Results of the cross validation experiments. Each point represents a test set. On the x-axis the average frequency of the test words in the training set is listed and the error rate obtained with this test set on the y-axis. Frequencies are given both for all words and for all unique words.

maining 18 sentences, 32 word errors occur (10 insertion errors, 10 deletion errors, and 12 substitution errors). Examples of remaining errors in the recognized sentences using the best presented automatic sign language recognition system are shown in Table 7.27. It can be seen that many errors occur in utterances of sentences that have not been seen in training, only one of these unseen sentences is recognized without any error. However, parts of the unknown sentences are recognized correctly showing that the visual models match also signs in an unknown context.

Deletions errors occur, for example, in the sentence "JOHN LIKE IX IX IX" (row 1) where "IX" is a pointing sign (meaning something like "John likes this subject, this subject, and this subject"). One of the pointing signs is recognized as part of the other signs. The unknown word in sentence 2 in Table 7.27 cannot be recognized correctly, because no visual model has been trained for it. The occurrence of unknown words can lead to subsequent faults, as happened in the recognition of this sentence. The word "JOHN" is inserted at the beginning of two sentences (rows 8 and 15). It has as a relatively high language model probability at the sentence begin, because of its frequent occurrence at this position in the training set (see Section 6.1). The insertion of the sign "MARY" (rows 14 and 15) is also caused by high language model probabilities. The main part of the sentence in row 14 is recognized correctly, even though it is performed by signer C during the training and by signer A in the test. Many of the correctly recognized sentences have been performed by several signers during the training (see Table A.5 in Appendix A), showing that the recognition is not person dependent.

Table 7.27. Examples of errors of the best presented system (see Table 7.23). The table shows the reference sentence aligned with the recognized sentence, the signer who performed the sentence and the distribution of signers who performed an utterance of the sentence in training. A list of all sentences can be found in Appendix A in Table A.5.

			samples		
row	reference, recognized	signer	A	B	C
1	JOHN LIKE IX IX IX JOHN LIKE IX IX ___	B	0	3	1
2	JOHN [UNKNOWN] BUY HOUSE JOHN FUTURE NOT BUY HOUSE	A	0	0	0
3	FUTURE JOHN BUY CAR SHOULD _________ JOHN BUY CAR FUTURE	A	0	0	0
4	JOHN SHOULD NOT BUY HOUSE JOHN FUTURE NOT BUY HOUSE	B	1	1	0
5	ANN BLAME MARY ANN BLAME ______	B	0	1	0
6	IX-1P FIND SOMETHING-ONE BOOK JOHN BUY WHAT YESTERDAY BOOK	A	0	0	0
7	JOHN IX GIVE MAN IX NEW COAT JOHN IX WOMAN _____ ___ NEW COAT	B	0	0	0
8	POSS NEW CAR BREAK-DOWN JOHN POSS NEW CAR BREAK-DOWN	A	1	0	0
9	JOHN LEG JOHN POSS LEG	A	0	0	0
10	JOHN POSS FRIEND HAVE CANDY JOHN POSS FRIEND _____ CANDY	C	0	0	0
11	IX CAR BLUE SUE BUY IX CAR BLUE ____ ____	C	0	0	1
12	JOHN READ BOOK JOHN FUTURE FINISH READ BOOK	B	0	1	0
13	JOHN IX SAY LOVE MARY JOHN IX SAY-1P LOVE MARY	C	0	0	0
14	JOHN ARRIVE JOHN APPLE WHO MARY	C	0	0	3
15	LIKE CHOCOLATE WHO JOHN LIKE CHOCOLATE WHO MARY	A	0	0	1
16	JOHN TELL MARY IX-1P BUY HOUSE JOHN FUTURE NOT _____ BUY HOUSE	B	0	0	0

The confusion matrix in Figure 7.18 shows that no significant accumulation of errors for a word or a combination of words arises. As mentioned in the previous paragraph, the word “IX” is deleted relatively often, the words “JOHN” and “MARY” are inserted into two sentences. “SHOULD” is confused with the sign “FUTURE” in both of its utterances. It can be seen that the number of insertion and deletion errors is balanced, meaning that the recognition system does neither recognize too many nor too few words.

	JOHN	WRITE	HOMEWORK	CAN	GO	FISH	WONT	EAT	BUT	CHICKEN	LIKE	IX	MARY	VEGETABLE	KNOW	CORN	THINK	LOVE	[UNKNOWN]	FUTURE	NOT	BUY	HOUSE	CAR	SHOULD	DECIDE	VISIT	WILL	ANN	BLAME	IX-1P	FIND	SOMETHING-ONE	WHAT	YESTERDAY	BOOK	GIVE	WOMAN	MAN	NEW	COAT	POSS	BREAK-DOWN	LEG	FRIEND	HAVE	CANDY	ARRIVE	BLUE	SUE	FINISH	READ	WHO	SAY	SAY-1P	PEOPLE	GROUP	JANA	TOY	APPLE	ALL	BOY	TEACHER	GIRL	BOX	CHOCOLATE	TELL	insertions	deletions
JOHN	30																																																																			2	
WRITE		1																																																																			
HOMEWORK			1																																																																		
CAN				5																																																																	
GO					2																																																																
FISH						1																																																															
WONT							1																																																														
EAT								2																																																													
BUT									1																																																												
CHICKEN										1																																																											
LIKE										5																																																										
IX											15																																																									3
MARY												7								1																																															2	1
VEGETABLE													1																																																							
KNOW														1																																																						
CORN															1																																																					
THINK																1																																																				
LOVE																	3																																																			
UNKNOWN																			1																																																	
FUTURE																			1																																																1	1
NOT																				3																																															1	
BUY																					8																																															1
HOUSE																						4																																														
CAR																							4																																													
SHOULD																			2																																																	
DECIDE																				1																																																
VISIT																										3																																										
WILL																											1																																									
ANN																												1																																								
BLAME																													2																																							
IX-1P	1																																																																				1
FIND																					1																																															
SOMETHING-ONE																																2	1																																			
WHAT																																	2																																			
YESTERDAY																																		2																																	1	
BOOK																																			6																																	
GIVE																																				6	1																															
WOMAN																																					3																															
MAN																																																																				1
NEW																																							2																													
COAT																																								1																												
POSS																																									2																										1	
BREAK-DOWN																																										1																										
LEG																																											1																									
FRIEND																																												1																								
HAVE																																																																				1
CANDY																																														1																						
ARRIVE																																															1												1									
BLUE																																																2																				
SUE																																																	1																			1
FINISH																																																																			1	
READ																																																			1																	
WHO																																																				2															1	
SAY																																																						1														
SAY-1P																																																																				
PEOPLE																																																							1													
GROUP																																																								1												
JANA																																																									1											
TOY																																																										1										
APPLE																																																											1									
ALL																																																												1								
BOY																																																													1							
TEACHER																																																														1						
GIRL																																																															2					
BOX																																																																2				
CHOCOLATE																																																																	1			
TELL																			1																																																	

Figure 7.18. Confusion matrix for the result of the best presented recognition system (see Table 7.23). The vertical axis shows the reference words, recognized words are on the horizontal axis.

Chapter 8

Conclusion and Perspectives

Conclusion

In this work a vision-based automatic sign language recognition was presented which is able to recognize sentences in American Sign Language. Several features and different methods to combine them were investigated and experimentally evaluated. Tracking algorithms with applications to hand and head tracking were presented and experiments were carried out to determine the parameters of these algorithms.

An emphasis was put on appearance-based features that use the images itself to represent signs. Other systems for automatic sign language recognition usually require a segmentation of input images to calculate features for the segmented image parts. The results presented in this work show that the usage of appearance-based features yields a promising recognition performance. It was shown that for a relative low amount of training data, principal component analysis is better suited to reduce the dimensionality of appearance-based features than linear discriminant analysis.

A set of features for manual components of signs was investigated in addition. These manual features were calculated using the output of the presented tracking algorithm. The usage of features describing the movement and the trajectory of the signer's dominant hand improved the recognition performance.

Moreover, different types of feature combination were investigated. Features combinations of one frame and of succeeding frames of the input video were analyzed. The experiments carried out reveal that the combination of succeeding feature vectors enhances the recognition results. Again, the results obtained with linear combinations of feature vectors using principal component analysis performed better than those obtained with linear discriminant analysis because of the limited number of training samples available for the calculation of the transformation matrices. The best recognition results were achieved by a combination of visual models for different types of combined features, each of them using both appearance-based and manual features.

The lengths of the signs were modeled by adapting the number of HMM states in the visual model of each sign according to the average sign length in the training data. It was necessary to include skip transitions in the visual models in order to recognize shorter utterances of a sign. During the training phase of the system, HMM topologies with skip transitions need a special consideration if only few training samples of signs are available.

The presented tracking algorithm was able to track the head and the dominant hand of the signers with a precision suitable for the calculation of manual and non-manual features. The tracking was formulated as optimization problem optimizing the complete path of object positions. This optimization problem was solved using dynamic programming. No initialization of the tracking or predefined models are needed. A method for hand tracking was developed that takes object and background appearance into account. The eigenface approach known from face detection was integrated in the tracking framework and used for the tracking of heads.

For the evaluation of hand tracking methods, a new database of sign language videos with annotated hand positions was created. Experiments with the sign language recognition system were carried out on a database consisting of 201 videos with annotated glosses. The videos show three persons performing sentences in American Sign Language. Although the database was originally created for linguistic research and thus contains only few utterances of many signs, a promising recognition performance was achieved.

The results obtained in this work show that models and methods commonly used in the field of speech recognition – like acoustic modeling with HMMs, language models, search techniques, and training algorithms – are applicable to sign language.

An important part of this work was the implementation of an automatic sign language recognition system in a speech recognition framework. The system offers a wide range of image processing methods and allows for a flexible combination of different features. The tracking algorithms were implemented as a part of the feature extraction module using the same software framework.

Perspective

The presented database includes a very small vocabulary only, because databases suitable for sign language recognition with a larger vocabulary are currently not available. Databases collected for linguistic research do usually not fulfill the requirements of a recognition system using a statistical approach, because they emphasize on structural variations and features of grammar without many repetitions of signs.

A challenging database is presented in [Bungeroth & Stein$^+$ 06]. This database consists of weather reports of the ARD Tagesschau translated to German Sign Language which have been telecasted at the German television channel Phoenix. Although the domain is limited to weather forecast, a large vocabulary is used but many signs occur only once. In addition, the translation is done by many interpreters, each of them speaking another dialect. Furthermore, the video material is challenging: The interpreter is shown in a relatively small part of the television frames with low resolution and changing, complex background. Furthermore, the signing speed is very high in order to keep the translation synchronous. At the moment, no meaningful results have been obtained with automatic sign language recognition using this database.

An important task is the creation of a standard test database for sign language recognition systems allowing the comparison of different systems in a quantitative way. The current situation, where most databases are not publicly available, prevents any objective comparison.

More features may be needed to distinguish between signs of a larger vocabulary. Especially features describing the hand shape need further investigations. If more than the lexical meaning of signs shall be recognized, the non-manual components of sign language have to be considered. For example, natural language understanding is necessary for dialog systems which requires more attributes of speech than recognized by the presented system. The translation of sign language to a spoken language needs knowledge about inflections and the recognition of subjects indexed by pointing gestures.

The presented tracking algorithms can be enhanced in different aspects. The tracking should allow for a variable object size. Therefore, the tracking window size can be optimized locally during the calculation of the scores, as proposed in [Dreuw 05], or over the complete path during the traceback. Another drawback of the presented tracking method is that only one object is tracked. A parallel tracking of several objects could be achieved by a refinement of the traceback. To apply the tracking algorithm to other tasks like human activity surveillance, appearing and disappearing objects have to be considered. Further improvements are expected if recognition and tracking are interleaved, resulting in only one decision to be taken for both tracking of the hands and recognition of a sign.

Appendix A

Additional Tables

Table A.5. Recognized sentences using the best presented recognition system (see Table 7.23). The table shows the reference sentence aligned with the recognized sentence, the signer who performed the sentence and the number of training samples for each signers.

reference, recognized	signer	samples		
		A	B	C
JOHN WRITE HOMEWORK JOHN WRITE HOMEWORK	A	1	0	0
JOHN CAN GO CAN JOHN CAN GO CAN	C	1	4	1
JOHN CAN GO CAN JOHN CAN GO CAN	B	1	4	1
JOHN FISH WONT EAT BUT CAN EAT CHICKEN JOHN FISH WONT EAT BUT CAN EAT CHICKEN	B	0	2	0
JOHN LIKE IX IX IX JOHN LIKE IX IX __	B	0	3	1
JOHN LIKE IX IX IX JOHN LIKE IX IX __	B	0	3	1
JOHN LIKE IX IX IX JOHN LIKE IX IX IX	C	0	3	1
MARY VEGETABLE KNOW IX LIKE CORN MARY VEGETABLE KNOW IX LIKE CORN	C	0	1	1
JOHN IX THINK MARY LOVE JOHN IX THINK MARY LOVE	C	0	1	1
JOHN [UNKNOWN] BUY HOUSE JOHN FUTURE NOT BUY HOUSE	A	0	0	0
FUTURE JOHN BUY CAR SHOULD _________ JOHN BUY CAR FUTURE	A	0	0	0

continued on the next page

		samples		
reference, recognized	signer	A	B	C
JOHN SHOULD NOT BUY HOUSE JOHN FUTURE NOT BUY HOUSE	B	1	1	0
JOHN DECIDE VISIT MARY JOHN NOT VISIT MARY	C	0	0	2
JOHN FUTURE NOT BUY HOUSE JOHN FUTURE NOT BUY HOUSE	A	3	0	0
JOHN WILL VISIT MARY JOHN WILL VISIT MARY	B	0	2	0
JOHN NOT VISIT MARY JOHN NOT VISIT MARY	C	0	0	1
ANN BLAME MARY ANN BLAME ______	B	0	1	0
IX-1P FIND SOMETHING-ONE BOOK JOHN BUY WHAT YESTERDAY BOOK	A	0	0	0
JOHN IX GIVE MAN IX NEW COAT JOHN IX WOMAN _____ __ NEW COAT	B	0	0	0
JOHN GIVE IX SOMETHING-ONE WOMAN BOOK JOHN GIVE IX SOMETHING-ONE WOMAN BOOK	C	0	1	1
JOHN GIVE IX SOMETHING-ONE WOMAN BOOK JOHN GIVE IX SOMETHING-ONE WOMAN BOOK	B	0	1	1
POSS NEW CAR BREAK-DOWN JOHN POSS NEW CAR BREAK-DOWN	A	1	0	0
JOHN LEG JOHN POSS LEG	A	0	0	0
JOHN POSS FRIEND HAVE CANDY JOHN POSS FRIEND _____ CANDY	C	0	0	0
WOMAN ARRIVE WOMAN ARRIVE	B	0	0	0
IX CAR BLUE SUE BUY IX CAR BLUE ____ ____	C	0	0	1
SUE BUY IX CAR BLUE SUE BUY IX CAR BLUE	C	0	0	2
JOHN READ BOOK JOHN FUTURE FINISH READ BOOK	B	0	1	0

continued on the next page

		samples		
reference, recognized	signer	A	B	C
JOHN BUY WHAT YESTERDAY BOOK JOHN BUY WHAT YESTERDAY BOOK	C	1	0	1
JOHN BUY YESTERDAY WHAT BOOK JOHN BUY YESTERDAY WHAT BOOK	A	1	0	1
LOVE JOHN WHO LOVE JOHN WHO	C	1	0	1
JOHN IX SAY LOVE MARY JOHN IX SAY-1P LOVE MARY	C	0	0	0
JOHN MARY BLAME JOHN MARY BLAME	C	0	0	1
PEOPLE GROUP GIVE JANA TOY PEOPLE GROUP GIVE JANA TOY	C	0	0	1
JOHN ARRIVE JOHN APPLE WHO MARY	C	0	0	3
ALL BOY GIVE TEACHER APPLE ALL BOY GIVE TEACHER APPLE	C	0	0	2
JOHN GIVE GIRL BOX JOHN GIVE GIRL BOX	C	0	0	2
JOHN GIVE GIRL BOX JOHN GIVE GIRL BOX	C	0	0	2
LIKE CHOCOLATE WHO JOHN LIKE CHOCOLATE WHO MARY	A	0	0	1
JOHN TELL MARY IX-1P BUY HOUSE JOHN FUTURE NOT _____ BUY HOUSE	B	0	0	0

Table A.1. Results for the features hand position u_t, hand motion m_t with $\Delta \in \{1, 2\}$, motion energy e_t and their combinations.

	m_t			
u_t	$\Delta = 1$	$\Delta = 2$	e_t	WER [%]
X				59.6
	X			56.7
		X		65.7
			X	72.5
	X	X		69.1
X	X			48.9
X		X		66.3
X	X	X		46.1
X			X	48.9
	X		X	59.6
		X	X	59.6
	X	X	X	65.7
X	X		X	44.4
X		X	X	46.1
X	X	X	X	42.1

Table A.2. Results for the combination of PCA-frames and hand position u_t, hand motion m_t with $\Delta \in \{1, 2\}$, motion energy e_t.

	m_t			
u_t	$\Delta = 1$	$\Delta = 2$	e_t	WER [%]
				27.5
X				25.3
	X			27.5
		X		27.5
			X	27.0
	X	X		24.2
X	X			28.7
X		X		27.0
X	X	X		27.0
X			X	27.5
	X		X	27.0
		X	X	27.0
	X	X	X	24.7
X	X	X	X	27.0

Table A.3. Results for trajectory features eigenvalues λ_1, λ_2, eigenvectors v_1, v_2 (sorted by decreasing eigenvalues), ratio of eigenvalues, and combinations with PCA-frames. The window size is set to $2\Delta + 1 = 5$

PCA-frame	λ_1,λ_2	v_1	v_2	$\frac{\lambda_1}{\lambda_2}$	WER [%]
	X				86.5
		X			73.6
		X	X		82.6
				X	86.5
X	X				23.6
X		X			26.4
X				X	25.8
X		X	X		27.0
X	X	X			26.4
X	X			X	25.8
X	X	X		X	27.5

Table A.4. Results for the combination of PCA-frames, manual features and the trajectory eigenvalue feature (window size 5).

	m_t			
u_t	$\Delta = 1$	$\Delta = 2$	e_t	WER [%]
X				28.1
	X			27.0
		X		24.7
			X	25.3
	X	X		27.0
X	X			24.7
X		X		28.1
X	X	X		26.4
X			X	25.3
	X		X	24.7
		X	X	26.4
	X	X	X	24.7
X	X	X	X	25.8

List of Figures

List of Tables

Bibliography

[Assan & Grobel 97] M. Assan, K. Grobel: Isolated Sign Language Recognition using Hidden Markov Models. In *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, pp. 162–167, Orlando, FL, USA, Oct. 1997.

[Bahl & Jelinek$^+$ 83] L.R. Bahl, F. Jelinek, R.L. Mercer: A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 179–190, March 1983.

[Baker 75] J.K. Baker: Stochastic Modeling for Automatic Speech Understanding. In D.R. Reddy, editor, *Speech Recognition*, pp. 512–542. Academic Press, New York, NY, USA, 1975.

[Baker & Padden 78] C. Baker, C.A. Padden: Focusing on the Nonmanual Components of ASL. In P.A. Siple, editor, *Understanding Language through Sign Language Research.*, Perspectives in Neurolinguistics and Psycholinguistics, pp. 27–57. Academic Press, New York, NY, USA, 1978.

[Bakis 76] R. Bakis: Continuous Speech Word Recognition via Centisecond Acoustic States. In *91st Meeting of the Acoustical Society of America (ASA)*, Washington, DC, USA, April 1976.

[Battison 78] R. Battison: *Lexical Borrowing in American Sign Language.* Linstok Press, Silver Spring, MD, USA, 1978.

[Bauer 04] B. Bauer: *Erkennung kontinuierlicher Gebärdensprache mit Untereinheiten-Modellen.* Shaker, Nov. 2004.

[Bauer & Hienz$^+$ 00] B. Bauer, H. Hienz, K.F. Kraiss: Video-Based Continuous Sign Language Recognition Using Statistical Methods. In *Proceedings of the International Conference on Pattern Recognition*, Vol. 2, pp. 463–466, Barcelona, Spain, Sept. 2000.

[Bauer & Kraiss 02] B. Bauer, K.F. Kraiss: Video-Based Sign Recognition using Self-Organizing Subunits. In *International Conference on Pattern Recognition (ICPR 2002)*, Vol. 2, pp. 434–437, Quėbec City, Canada, Aug. 2002.

[Baum 72] L.E. Baum: An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes. In O. Shisha, editor, *Inequalities*, Vol. 3, pp. 1–8. Academic Press, New York, NY, USA, 1972.

[Bayes 63] T. Bayes: An Essay towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London*, Vol. 53, pp. 370–418, 1763.

[Bellmann 57] Bellmann: *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA, 1957.

[Bellugi & Fischer 72] U. Bellugi, S. Fischer: A Comparison of Sign Language and Spoken Language. *Cognition*, Vol. 1, No. 2–3, pp. 173–200, 1972.

[Bobick & Wilson 97] A.F. Bobick, A.D. Wilson: A State-Based Approach to the Representation and Recognition of Gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 12, pp. 1325–1337, 1997.

[Bowden & Windridge$^+$ 04] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, M. Brady: A Linguistic Feature Vector for the Visual Interpretation of Sign Language. In *8th European Conference on Computer Vision (ECCV), Part IV*, Vol. 3024 of *Lecture Notes in Computer Science*, pp. 390–401, Prague, Czech Republic, May 2004. Springer.

[Bradski 98] G.R. Bradski: Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal*, Vol. 2, No. 2, pp. 15–26, 1998.

[Braem 95] P.B. Braem: *Einführung in die Gebärdensprache und ihre Erforschung*, Vol. 11 of *Internationale Arbeiten zur Gebärdensprache und Kommunikation Gehörloser.* Signum, Hamburg, Germany, 3rd edition, 1995.

[Bridle & Sedgwick 77] J.S. Bridle, N.C. Sedgwick: A Method for Segmenting Acoustic Patterns, with Applications to Automatic Speech Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '77)*, Vol. 2, pp. 656–659, May 1977.

[Bungeroth & Ney 04] J. Bungeroth, H. Ney: Statistical Sign Language Translation. In *Workshop on Representation and Processing of Sign Languages, 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 105–108, Lisbon, Portugal, May 2004.

[Bungeroth & Stein$^+$ 06] J. Bungeroth, D. Stein, P. Dreuw, M. Zahedi, H. Ney: A German Sign Language Corpus of the Domain Weather Report. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May 2006.

[Canzler 05] U. Canzler: *Nicht-intrusive Mimikanalyse.* Ph.D. thesis, RWTH Aachen University, Lehrstuhl für technische Informatik, Aachen, Germany, 2005.

[Comaniciu & Ramesh$^+$ 00] D. Comaniciu, V. Ramesh, P. Meer: Real-Time Tracking of Non-Rigid Objects using Mean Shift. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, Vol. 2, pp. 142–151, Hilton Head Island, SC, USA, June 2000.

[Dreuw 05] P. Dreuw: Appearance-Based Gesture Recognition. Diploma thesis, RWTH Aachen University, Aachen, Germany, Jan. 2005.

[Dreuw & Deselaers$^+$ 06] P. Dreuw, T. Deselaers, D. Rybach, D. Keysers, H. Ney: Tracking Using Dynamic Programming for Appearance-Based Sign Language Recognition. In *7th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2006)*, pp. 293–298, Southampton, UK, April 2006.

[Dreuw & Keysers$^+$ 06] P. Dreuw, D. Keysers, T. Deselaers, H. Ney: Modeling Image Variability in Appearance-Based Gesture Recognition. In *3rd Workshop on Statistical Methods in Multi-Image and Video Processing (ECCV-SMVP)*, Graz, Austria, May 2006.

[Duda & Hart$^+$ 01] R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification.* John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.

[Elliott & Glauert$^+$ 00] R. Elliott, J.R.W. Glauert, J.R. Kennaway, I. Marshall: The Development of Language Processing Support for the ViSiCAST Project. In *4th International ACM Conference on Assistive Technologies*, pp. 101 – 108, Washington, DC, USA, Nov. 2000.

[Fang & Gao 02] G. Fang, W. Gao: A SRN/HMM System for Signer-Independent Continuous Sign Language Recognition. In *5th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2002)*, pp. 312–317, Washington, DC, USA, May 2002.

[Fisher 04] R.B. Fisher: The PETS04 Surveillance Ground-Truth Data Sets. In *6th International Workshop on Performance Evaluation for Tracking and Surveillance*, pp. 1–5, Prague, Czech Republic, May 2004.

[Gauvain & Lamel$^+$ 05] J. Gauvain, L. Lamel, H. Schwenk, F. Brugnara, R. Schlüter, M. Bisani, S. Stüker, T. Schaaf, S. Mohammed, M. Bacchiani, M. Westphal, S. Sivadas, I. Kiss, F. Giron: ASR Progress Report. Technical report, Technology and Corpora for Speech to Speech Translation (TC-STAR), May 2005.

[Gavrila 99] D.M. Gavrila: The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, Vol. 73, No. 1, pp. 82–98, Feb. 1999.

[Generet & Ney$^+$ 95] M. Generet, H. Ney, F. Wessel: Extensions to Absolute Discounting for Language Modeling. In *4th European Conference on Speech Communication and Technology (EUROSPEECH 95)*, Vol. 2, pp. 1245–1248, Madrid, Spain, Sept. 1995.

[Haeb-Umbach & Ney 92] R. Haeb-Umbach, H. Ney: Linear Discriminant Analysis for Improved Large Vacabulary Continuous Speech Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '92)*, Vol. 1, pp. 13–16, San Francisco, CA, USA, March 1992.

[Hastie & Tibshirani$^+$ 03] T. Hastie, R. Tibshirani, J.H. Friedman: *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, 2003.

[Holden & Lee$^+$ 05] E.J. Holden, G. Lee, R. Owens: Automatic Recognition of Colloquial Australian Sign Language. In *IEEE Workshop on Motion and Video Computing (WACV/MOTION '05)*, Vol. 2, pp. 183–188, Orlando, FL, USA, Dec. 2005.

[Holden & Owens 00] E.J. Holden, R. Owens: Visual Sign Language Recognition. In *10th International Workshop on Theoretical Foundations of Computer Vision*, Vol. 2032 of *Lecture Notes in Computer Science*, pp. 270–288, Dagstuhl Castle, Germany, March 2000. Springer.

[Huang & Huang 98] C.L. Huang, W.Y. Huang: Sign Language Recognition using Model-based Tracking and a 3D Hopfield Neural Network. *Machine Vision and Applications*, Vol. 10, No. 5-6, pp. 292–307, April 1998.

[Huang & Sebe$^+$ 06] T. Huang, N. Sebe, M. Lew, V. Pavlovic, M. Kölsch, A. Galata, B. Kisacanin, editors: *Workshop on Human-Computer Interaction*, Vol. 3979 of *Lecture Notes in Computer Science*. Springer, May 2006.

[Hwang & Hon$^+$ 89] M. Hwang, H. Hon, K. Lee: Modeling Between-Word Coarticulation in Continuous Speech Recognition. In *European Conference on Speech Technology (EUROSPEECH 89)*, pp. 5–8, Paris, France, Sept. 1989.

[Isard & Blake 98] M. Isard, A. Blake: CONDENSATION – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, Vol. 29, No. 1, pp. 5–28, Aug. 1998.

[Jelinek 98] F. Jelinek: *Statistical Methods for Speech Recognition.* MIT Press, Jan. 1998.

[Jolliffe 02] I. Jolliffe: *Principal Component Analysis.* Springer, 2nd edition, 2002.

[Jones & Rehg 02] M.J. Jones, J.M. Rehg: Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision*, Vol. 46, No. 1, pp. 81–96, Jan. 2002.

[Kanthak & Molau$^+$ 00] S. Kanthak, S. Molau, A. Sixtus, R. Schlüter, H. Ney: RWTH Large Vocabulary Speech Recognition System for Spontaneous Speech. In *5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS)*, pp. 249–254, Ilmenau, Germany, Oct. 2000.

[Katz 87] S.M. Katz: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 35, pp. 400–401, March 1987.

[Katz & Meier$^+$ 02] M. Katz, H.G. Meier, H. Dolfing, D. Klakow: Robustness of Linear Discriminant Analysis in Automatic Speech Recognition. In *International Conference on Pattern Recognition (ICPR 2002)*, Vol. 3, pp. 371–374, Québec City, Canada, Aug. 2002.

[Keysers & Macherey$^+$ 01] D. Keysers, W. Macherey, J. Dahmen, H. Ney: Learning of Variability for Invariant Statistical Pattern Recognition. In *12th European Conference on Machine Learning (ECML '01)*, Vol. 2167 of *Lecture Notes in Computer Science*, pp. 263–275, Freiburg, Germany, Sept. 2001. Springer.

[Keysers & Macherey$^+$ 04] D. Keysers, W. Macherey, H. Ney, J. Dahmen: Adaptation in Statistical Pattern Recognition using Tangent Vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 2, pp. 269–274, Feb. 2004.

[Klima & Bellugi 79] E.S. Klima, U. Bellugi: *The Signs of Language.* Harvard University Press, Cambridge, UK, 1979.

[Levenshtein 66] V.I. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10, pp. 707–710, 1966.

[Liddell & Johnson 89] S. Liddell, R. Johnson: S. Liddell, R. Johnson: American Sign Language: The Phonological Base. *Sign Language Studies*, Vol. 64, pp. 195–277, Jan. 1989.

[Linde & Buzo$^+$ 80] Y. Linde, A. Buzo, R. Gray: An Algorithm for Vector Quantization Design. In *IEEE Transactions on Communications*, Vol. 28, pp. 84–95, Jan. 1980.

[Martinez & Kak 01] A. Martinez, A. Kak: PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, pp. 228–233, Feb. 2001.

[Neidle 01] C. Neidle: SignStream$^{\text{TM}}$: A Database Tool for Research on Visual-Gestural Language. *Sign Language and Linguistics*, Vol. 4, No. 1/2, pp. 203–214, 2001.

[Neidle & Kegl$^+$ 99] C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, R. Lee: *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure.* MIT Press, Cambridge, MA, USA, Dec. 1999.

[Ney 84] H. Ney: The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Transactions on Speech and Audio Processing*, Vol. 32, No. 2, pp. 263–271, April 1984.

[Ney 90] H. Ney: Acoustic Modeling of Phoneme Units for Continuous Speech Recognition. In *5th European Signal Processing Conference, Signal Processing V: Theories and Applications*, pp. 65–72, 65-72, Dec. 1990. Elsevier Science Publishers.

[Ney 99] H. Ney: Speech Translation: Coupling of Recognition and Translation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, Vol. 1, pp. 517–520, Phoenix, AZ, USA, March 1999.

[Ney 05a] H. Ney: One Decade of Statistical Machine Translation. In *MT Summit X*, pp. i–12 – i–17, Phuket, Thailand, Sept. 2005.

[Ney 05b] H. Ney: Speech Recognition. Script to the Lecture on Speech Recognition Held at RWTH Aachen University, 2005.

[Ney & Essen$^+$ 94] H. Ney, U. Essen, R. Kneser: On Structuring Probabilistic Dependencies in Language Modeling. *Computer Speech and Language*, Vol. 2, No. 8, pp. 1–38, 1994.

[Ney & Martin$^+$ 97] H. Ney, S.C. Martin, F. Wessel: Statistical Language Modeling using Leaving-One-Out. In *Corpus Based Methods in Language and Speech Processing*, chapter 6, pp. 174–207. Kluwer Academic Publishers, Dordrecht, Netherlands, Feb. 1997.

[Ney & Mergel$^+$ 87] H. Ney, D. Mergel, A. Noll, A. Paeseler: A Data-Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '87)*, pp. 833–836, Dallas, TX, USA, April 1987.

[Ong & Ranganath 05] S.C. Ong, S. Ranganath: Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp. 873–891, June 2005.

[Pavlovic & Sharma$^+$ 97] V. Pavlovic, R. Sharma, T.S. Huang: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677–695, July 1997.

[Poizner & Klima$^+$ 83] H. Poizner, E. Klima, U. Bellugi, R. Livingston: Motion Analysis of Grammatical Processes in a Visual-Gestural Language. In *ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion, Representation and Perception*, pp. 271–292, Ontario, Canada, 1983.

[Prillwitz & Leven$^+$ 89] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, J. Henning: *HamNoSys Version 2.0. Hamburg Notation System for Sign Languages. An introductory guide*, Vol. 5 of *International Studies on Sign Language and Communication of the Deaf.* Signum, Hamburg, Germany, 1989.

[Rabiner 89] L.R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, Vol. 77, pp. 257–286, Feb. 1989.

[Rabiner & Juang 86] L. Rabiner, B.H. Juang: An Introduction to Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 4–16, 1986.

[Rowley & Baluja$^+$ 98] H.A. Rowley, S. Baluja, T. Kanade: Neural Network-Based Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 23–38, Jan. 1998.

[Sandler 99] W. Sandler: Prosody in Two Natural Language Modalities. *Language and Speech*, Vol. 42, No. 2–3, pp. 127–142, April 1999.

[Sandler 03] W. Sandler: Sign Language Phonology. In W.J. Frawley, editor, *International Encyclopedia of Linguistics.* Oxford University Press, 2nd edition, 2003.

[Schlenzig & Hunter$^+$ 94] J. Schlenzig, E. Hunter, R. Jain: Recursive Identification of Gesture Inputs using Hidden Markov Models. In *2nd Annual Conference on Applications of Computer Vision*, pp. 187–194, Sarasota, FL, USA, Dec. 1994.

[Simard & LeCun$^+$ 98] P. Simard, Y. LeCun, J.S. Denker, B. Victorri: Transformation Invariance in Pattern Recognition – Tangent Distance and Tangent Propagation. In G.B. Orr, K.R. Müller, editors, *Neural Networks: Tricks of the Trade*, Vol. 1524 of *Lecture Notes In Computer Science*, pp. 239–274. Springer, 1998.

[Starner & Pentland 94] T. Starner, A. Pentland: Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pp. 84–91, June 1994.

[Starner & Weaver$^+$ 98] T. Starner, J. Weaver, A. Pentland: Real-time American Sign Language Recognition using Desk and Wearable Computer Based Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, pp. 1371–1375, Dec. 1998.

[Stein 05] D. Stein: Morpho-Syntax Based Statistical Methods for Sign Language Translation. Diploma thesis, RWTH Aachen University, Aachen, Germany, Nov. 2005.

[Stein & Bungeroth$^+$ 06] D. Stein, J. Bungeroth, H. Ney: Morpho-Syntax Based Statistical Methods for Sign Language Translation. In *11th Annual Conference of the European Association for Machine Translation*, Oslo, Norway, June 2006.

[Stokoe 60] W. Stokoe: Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf. *Studies in Linguistics, Occasional Papers*, Vol. 8, pp. 1–78, 1960.

[Stokoe & Casterline$^+$ 65] W. Stokoe, D. Casterline, C. Croneberg: *A Dictionary of American Sign Language on Linguistic Principles.* Gallaudet College Press, Washington D.C., USA, 1965.

[Stolcke 02] A. Stolcke: SRILM – An Extensible Language Modeling Toolkit. In *International Conference on Spoken Language Processing (ICSLP 2002)*, Vol. 2, pp. 901–904, Denver, CO, USA, Sept. 2002.

[Tanibata & Shimada$^+$ 02] N. Tanibata, N. Shimada, Y. Shirai: Extraction of Hand Features for Recognition of Sign Language Words. In *15th International Conference on Vision Interface (VI 2002)*, pp. 391–398, Calgary, Canada, May 2002.

[Turk & Pentland 91] M. Turk, A. Pentland: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71–86, 1991.

[Viola & Jones 04] P. Viola, M. Jones: Robust Real-Time Face Detection. *International Journal of Computer Vision*, Vol. 57, No. 2, pp. 137–154, 2004.

[Viterbi 67] A.J. Viterbi: Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, Vol. 13, No. 2, pp. 260–269, April 1967.

[Vogler & Metaxas 99a] C. Vogler, D. Metaxas: Parallel Hidden Markov Models for American Sign Language Recognition. In *7th IEEE International Conference on Computer Vision (ICCV 1999)*, Vol. 1, pp. 116–122, Kerkyra, Greece, Sept. 1999.

[Vogler & Metaxas 99b] C. Vogler, D. Metaxas: Toward Scalability in ASL Recognition: Breaking down Signs into Phonemes. In *International Gesture Workshop (GW'99)*, Vol. 1739 of *Lecture Notes in Artificial Intelligence*, pp. 211–224, Gif-sur-Yvette, France, March 1999. Springer.

[Vogler & Metaxas 01] C. Vogler, D. Metaxas: A Framework for Recognizing the Simultaneous Aspects of American Sign Language. *Computer Vision and Image Understanding*, Vol. 81, No. 3, pp. 358–384, March 2001.

[von Agris & Schneider^{+} 06] U. von Agris, D. Schneider, J. Zieren, K.F. Kraiss: Signer Adaptation for Isolated Sign Language Recognition. In *IEEE Workshop on Vision for Human Computer Interaction (V4HCI)*, New York, NY, USA, June 2006.

[Wessel & Ortmanns^{+} 97] F. Wessel, S. Ortmanns, H. Ney: Implementation of Word Based Statistical Language Models. In *SQEL Workshop on Multi-Lingual Information Retrieval Dialogs*, pp. 55–59, Pilsen, Czech Republic, April 1997.

[Woodland 01] P.C. Woodland: Speaker Adaptation for Continuous Density HMMs: A Review. In *ISCA ITR-Workshop on Adaptation Methods for Speech Recognition*, pp. 11–19, Sophia Antipolis, France, Aug. 2001.

[Wrobel 01] U.R. Wrobel: Referenz in Gebärdensprachen: Raum und Person. *Forschungsberichte des Instituts für Phonetik und Sprachliche Kommunikation der Universität München*, Vol. 37, pp. 25–50, 2001.

[Yang & Ahuja^{+} 02] M.H. Yang, N. Ahuja, M. Tabb: Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 8, pp. 1061–1074, Aug. 2002.

[Zahedi & Dreuw^{+} 06] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, H. Ney: Continuous Sign Language Recognition - Approaches from Speech Recognition and Available Data Resources. In *2nd Workshop on the Representation and Processing of Sign Languages: Lexicographic Matters and Didactic Scenarios*, Genoa, Italy, May 2006.

[Zahedi & Keysers^{+} 05] M. Zahedi, D. Keysers, H. Ney: Pronunciation Clustering and Modeling of Variability for Appearance-Based Sign Language Recognition. In *6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France, May 2005.

[Zhang & Chen^{+} 04] L.G. Zhang, Y. Chen, G. Fang, X. Chen, W. Gao: A Vision-Based Sign Language Recognition System using Tied-Mixture Density HMM. In *6th International Conference on Multimodal Interfaces (ICMI '04)*, pp. 198–204, New York, NY, USA, 2004. ACM Press.

[Zieren & Kraiss 05] J. Zieren, K.F. Kraiss: Robust Person-Independent Visual Sign Language Recognition. In *2nd Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2005)*, Vol. 3522 of *Lecture Notes in Computer Science*, pp. 520–528, Estoril, Portugal, June 2005. Springer.

[Zolnay & Schlüter+ 05] A. Zolnay, R. Schlüter, H. Ney: Acoustic Feature Combination for Robust Speech Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, Vol. 1, pp. 457–460, Philadelphia, PA, USA, March 2005.