Faculty of Applied Sciences

Department of Cybernetics

Doctoral thesis
submitted for the degree Doctor of Philosophy
in the field of Cybernetics

Ing. Pavel Campr

Automatic Sign Language Recognition from Image Data

Advisor: Doc. Ing. Miloš Železný, Ph.D.

Pilsen, 2012

Fakulta aplikovaných věd

Katedra kybernetiky

Disertační práce
k získání akademického titulu doktor
v oboru Kybernetika

Ing. Pavel Campr

Automatické rozpoznávání znakového jazyka z obrazových dat

Školitel: Doc. Ing. Miloš Železný, Ph.D.

Plzeň, 2012

Abstract

This thesis addresses several issues of automatic sign language recognition, namely the creation of vision based sign language recognition framework, sign language corpora creation, feature extraction, making use of novel hand tracking with face occlusion handling, data-driven creation of sub-units and "search by example" tool for searching in sign language corpora using hand images as a search query.

The proposed sign language recognition framework, based on statistical approach incorporating hidden Markov models (HMM), consists of video analysis, sign modeling and decoding modules. The framework is able to recognize both isolated signs and continuous utterances from video data.

All experiments and evaluations were performed on two own corpora, UWB-06-SLR-A and UWB-07-SLR-P, the first containing 25 signs and second 378.

As a baseline feature descriptors, low level image features are used. It is shown that better performance is gained by higher level features that employ hand tracking, which resolve occlusions of hands and face. As a side effect, the occlusion handling method interpolates face area in the frames during the occlusion and allows to use face feature descriptors that fail in such a case, for instance features extracted from active appearance models (AAM) tracker. Several state-of-the-art appearance-based feature descriptors were compared for tracked hands, such as local binary patterns (LBP), histogram of oriented gradients (HOG), high-level linguistic features or newly proposed hand shape radial distance function (denoted as hRDF) that enhances the feature description of hand-shape like concave regions.

The concept of sub-units, that uses HMM models based on linguistic units smaller than whole sign and covers inner structures of the signs, was investigated in the proposed iterative method that is a first required step for data-driven construction of sub-units, and shows that such a concept is suitable for sign modeling and recognition tasks.

Except of experiments in the sign language recognition, additional tool *search by example* was created and evaluated. This tool is a search engine for sign language videos. Such a system can be incorporated into an online sign language dictionary where it is difficult to search in the sign language data. This proposed tool employs several methods which were examined in the sign language recognition task and allows to search in the video corpora based on an user-given query that consists of one or multiple images of hands. As a result, an ordered list of videos that contain the same or similar hand configurations is returned.

Abstrakt

Tato práce se zabývá problematikou automatického rozpoznávání znakového jazyka z obrazových dat. Práce představuje pět hlavních přínosů v oblasti tvorby systému pro rozpoznávání, tvorby korpusů, extrakci příznaků z rukou a obličeje s využitím metod pro sledování pozice a pohybu rukou (tracking) a modelování znaků s využitím menších fonetických jednotek (subunits). Metody využité v rozpoznávacím systému byly využity i k tvorbě vyhledávacího nástroje "search by example", který dokáže vyhledávat ve videozáznamech podle obrázku ruky.

Navržený systém pro automatické rozpoznávání znakového jazyka je založen na statistickém přístupu s využitím skrytých Markovových modelů, obsahuje moduly pro analýzu video dat, modelování znaků a dekódování. Systém je schopen rozpoznávat jak izolované, tak spojité promluvy.

Veškeré experimenty a vyhodnocení byly provedeny s vlastními korpusy UWB-06-SLR-A a UWB-07-SLR-P, první z nich obsahuje 25 znaků, druhý 378.

Základní extrakce příznaků z video dat byla provedena na nízkoúrovňových popisech obrazu. Lepších výsledků bylo dosaženo s příznaky získaných z popisů vyšší úrovně porozumění obsahu v obraze, které využívají sledování pozice rukou a metodu pro segmentaci rukou v době překryvu s obličejem. Navíc, využitá metoda dokáže interpolovat obrazy s obličejem v době překryvu a umožňuje tak využít metody pro extrakci příznaků z obličeje, které by během překryvu nefungovaly, jako např. metoda active appearance models (AAM). Bylo porovnáno několik různých metod pro extrakci příznaků z rukou, jako např. local binary patterns (LBP), histogram of oriented gradients (HOG), vysokoúrovnové lingvistické příznaky a nové navržená metoda hand shape radial distance function (hRDF).

Bylo také zkoumáno využití menších fonetických jednotek, než jsou celé znaky, tzv. subunits. Pro první krok tvorby těchto jednotek byl navržen iterativní algoritmus, který tyto jednotky automaticky vytváří analýzou existujících dat. Bylo ukázáno, že tento koncept je vhodný pro modelování a rozpoznávání znaků.

Kromě systému pro rozpoznávání je v práci navržen a představen systém "search by example", který funguje jako vyhledávací systém pro videa se záznamy znakového jazyka a může být využit například v online slovnících znakového jazyka, kde je v současné době složité či nemožné v takovýchto datech vyhledávat. Tento nástroj využívá metody, které byly použity v rozpoznávacím systému. Výstupem tohoto vyhledávacího nástroje je seřazený seznam videí, které obsahují stejný nebo podobný tvar ruky, které zadal uživatel, např. přes webkameru.

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně, s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne Pavel Campr

Acknowledgements

I would like to thank to my advisor Doc. Ing. Železný Miloš, Ph.D., former advisor Doc. Ing. Müller Luděk, Ph.D. and head of my department Prof. Ing. Psutka Josef, CSc. for their support and patience throughout the time I was working on this thesis.

Furthermore, I would like to thank to the other colleagues of our Department of Cybernetics, both past and present, for their help, support and fruitful discussions. Special acknowledgements go to the participants of the eNTERFACE workshops that I worked together with and to the staff at the university who make things work.

I record my eternal thanks to my family and friends, who have been as understanding and encouraging as any doctoral student could wish for.

The financial assistance of the Grant Agency of Academy of Sciences of the Czech Republic (project No. 1ET101470416), of the Ministry of Education of the Czech Republic (project No. ME08106), of the EU and the Ministry of Education of the Czech Republic (project No. CZ.1.07/2.2.00/07.0189) and of the Grant Agency of the Czech Republic (project No. P103/12/G084) is fully acknowledged.

The access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the project LM2010005 (*Projects of Large Infrastructure for Research, Development, and Innovations*) is highly appreciated.

Contents

List of Figures

List of Tables

Glossary

AAM	active appearance model
acc	accuracy
ASL	American Sign Language
ASR	automatic speech recognition
ASM	active shape model
BSL	British Sign Language
CI	confidence interval
corr	correctness
CzSL	Czech Sign Language
Deaf (person)	a person who identifies with Deaf culture
DCT	discrete cosine transform
gesture	human-made hand movement intended for communication
Gaussian	mixture component of GMM
GMM	Gaussian mixture model
EM (algorithm)	expectation-maximization (algorithm)
HamNoSys	Hamburg Notational System, notation system for transcription of a sign language
HCI	human-computer interface
HOG	histogram of oriented gradients
HMM	hidden Markov model
hRDF	radial distance function, with modification proposed in this work
HTK	Hidden Markov model toolkit

LBP	local binary patterns
LDA	linear discriminant analysis
manual component	aspect of sign languages consisting of hand actions
non-manual component	aspect of sign languages that goes beyond the hand actions, especially face expression
PCA	principal component analysis
px	pixel
RDF	radial distance function
RGB	red green blue, a color space
sign	morphological item of sign languages, corresponding to *words* of spoken languages
SIGN GLOSS	notation of the meaning of a sign in written form of oral language
SignWriting	notation system for transcription of a sign language
SL	(any) sign language
SLR, ASLR	(automatic) sign language recognition
SVM	support vector machine
sub-unit	phoneme unit smaller than word
UWB	University of West Bohemia (Západočeská univerzita)
WER	word error rate

Symbols

(x, y)	coordinate in a 2D plane or an image
$[-1, 0, 1]$	a vector
$\boldsymbol{A} = \lvert a_{ij} \rvert$	matrix $\boldsymbol{A}$ with elements a_{ij}, with row index i and column index j
Δ	velocity parameters used in HMM modeling
Δ^2	acceleration and velocity parameters used in HMM modeling
λ_i	i-th eigenvalue
μ	mean value
$\boldsymbol{O} = \boldsymbol{o}_0, \boldsymbol{o}_1, \ldots$	observation represented by a *feature vector*
P	probability
$P(W)$	a priori probability of the given word sequence W
$p(\boldsymbol{O}\lvert W)$	likelihood of the observation sequence given the word sequence W
R^d	d-dimensional Euclidean space
t	time
T	period of time
$\boldsymbol{W} = \boldsymbol{w}_0, \boldsymbol{w}_1, \ldots$	sequence of words or signs

1 Introduction

Sign languages are used worldwide as a primary means of communication by deaf people. With the development of camera hardware in last decade and its availability, there are demands for research in the field of Human Computer Interaction (HCI) enabling the sign language to be a natural mean of communication between the computers and humans. The results of the research can be used in other fields such as linguistics or education. One of the main challenges is to develop an automatic system that can serve as an interpreter between sign and spoken languages, to allow communication of deaf people with others who use spoken language. This is not feasible nowadays, due to a still unresolved subtask of sign language recognition (SLR) that tries to recognize signs in a stream of video data.

The field of SLR faces similar problems as automatic speech recognition (ASR) did in its 50 year history. Thus it can profitably utilize methods and paradigms, where a tremendous research effort has been already put in. Despite this, the developments in SLR are many years behind. The reasons are the lack of available data, small number of researchers and small target group.

Like the spoken languages, sign languages evolve naturally within the deaf communities, independently from the spoken language of the region or country. Thus, each sign language has own grammar and lexicon, but share a common property that they are conveyed through multiple visual communication channels, incorporating mainly hands and head. This makes the analysis of sign language more complex task in comparison to one dimensional audio signal in speech. This visual manner of communication incorporates different language concepts than which are used in spoken languages.

In general, for most of the sign languages, no written form of the language is established and widely used. These days, instead of writing, the sign languages can be recorded, archived and sent in a digital video form. Another practical use case for the SLR research is to allow searching in this type of video data, that contains sign language recordings.

Sign language recognition is a multidisciplinary area that involves pattern recognition, computer vision, linguistics and natural language processing. Initially, cumbersome data gloves were used for data collection, but for real world applications this has been replaced by contactless camera-based devices, now even embedded in mobile devices.

In this thesis, several aspects of sign language recognition field are studied, such as hand tracking, particularly during an occlusion with the face, hand feature extraction, face feature extraction, sign modeling, recognition both of isolated signs and continuous speech, and data-driven analysis of smaller phoneme units. A SLR system was built and its performance was

measured on two corpora, UWB-06-SLR-A and UWB-07-SLR-P that were recorded for the purpose of this thesis. Although the corpora include isolated signs only, artificial continuous utterances were generated to perform continuous recognition. Multiple feature extraction methods were examined, some were based on low level image information, other were using higher level information about hand and head positions together with hand segmentation. Beyond the SLR system, a *search by example* system was proposed and evaluated. The system allows searching in sign language videos having one or multiple hand images as a search query.

1.1 Structure of This Document

This document is organized as follows. Chapter 2 introduces scope and scientific goals, addressing the main contributions of this work.

Chapter 3 contains a deeper introduction to sign languages, their notation systems and relations with Human Computer Interaction (HCI), especially the aspects of sign languages that are crucial for building of an automatic sign language recognition (SLR) system or other HCI applications.

Chapter 4 summarizes available datasets, including the description of our recorded corpora. The datasets are compared and their suitability for SLR is discussed, due to the fact that most of the datasets were recorded for usage in linguistics and not HCI research.

Chapter 5 is an overview of current approaches and methods that are employed in SLR. Most systems use statistical approach, employing Bayes decision rule and hidden Markov models. The recognition system consists of several modules. The first module performs analysis of a source video, resulting in a series of feature vectors. Multiple approaches, mostly based on appearance-based modeling, processing both manual (hands) and non-manual (face) components of a sign, are discussed. Furthermore, supportive methods such as feature decorrelation and dimension reduction are included. Next section, *sign modeling*, is a brief introduction to hidden Markov models and its application in the field of SLR. *Language Modeling* section introduces possibilities for the use of language models for sign languages. Finally, the *sign decoding* module is described, involving possibilities of quality evaluation.

The main chapter 6 reveals contributions and results of this work, particularly in video analysis and sign modeling. Multiple approaches for sign language recognition task are introduced and their performance is compared. Then, the *search by example* system is introduced and evaluated.

The last chapter 7 concludes the document. The achieved results are summarized and future perspectives of the sign language recognition field are given.

2 | Scope and Goals of the Thesis

The main goal of the thesis is to build a sign language recognition system and establish a basis for further experiments and evaluations. Such a system consists of several modules, each rising its own problems. The particular goals are to improve feature extraction and sign modeling methods. Two different real-world applications are considered for evaluation: **1) sign language recognition (SLR) system**, both for recognition of isolated signs and for continuous utterances; **2) search by example system**, which allows searching in sign video corpora, where single or multiple images containing hands in particular configurations are given as a search query.

2.1 Goals of the Thesis

1. To develop a baseline SLR system to allow further research in the area, both for recognition of isolated signs and continuous utterances.
2. To record a dataset suitable for experiments and evaluations of the SLR system.
3. To compare existing feature extraction methods that analyze input frames recorded by a stationary camera. Both low level image features and higher level appearance-based features, incorporating hand and head tracking, are considered.
4. To improve particular feature extraction techniques, with focus on hand tracking and occlusion handling.
5. To evaluate recognition performance for different parameters of the SLR system, for different features and their modifications, such as dimension reduction and fusion.
6. To investigate the construction of smaller phoneme units, *sub-units*, by unsupervised analysis of the used datasets, and evaluate its usage in the SLR system.
7. To use the integration of proposed methods to build a *search by example* system, that uses images containing hands as a search query. The motivation behind is to embed such a search system into an online sign language dictionary application, so that the user is allowed to search the video content by posting queries that consist of hand images captured by a webcam.

3 | Sign Languages

Sign languages use manual communication means and body language to convey meaning, instead of acoustic sound patterns used in spoken languages. The speaker, *signer*, uses combination of hand shapes, orientation and movements of the hands, arms and body, and facial expressions to communicate.

There are around two hundred sign languages around the world. They develop wherever communities of deaf people exist.

Deaf sign languages are preferred by the deaf, and they are a natural form of sign languages developed in a community of deaf people.

Signed modes of spoken languages (manually coded languages) are a bridge between spoken and sign languages. Likewise, they use signs, but the grammar and sentence structure are adopted from spoken languages.

Auxiliary sign systems are artificial signed systems, sometimes used together with spoken languages, for instance: International Sign (an auxiliary language used by the deaf in international settings), Baby Sign (used in early language development in young children), Military hand and arm signals, Tic-tac (used by bookmakers to communicate the odds at racecourses).

Cued speech is a phonemic-based system. It makes traditionally spoken languages accessible by using a small number of handshapes (representing consonants) in different locations near the mouth (representing vowels), as a supplement to speechreading [wik12].

It is quite difficult to estimate number of people who are deaf and who use sign languages. In the Czech Republic the number is about 7500 of deaf signers (0.07% of the population) [Hru09], and about 500000 (4.7%) hard of hearing. In Germany, the estimation is about 100000 (0.12%) of people using sign languages [FSH+12].

Sign languages are now recognized as fully legitimate languages in countries throughout the world.

There is one major difference between sign and spoken languages. In most of the sign languages, signs can be meaningfully placed or directed in space. One instance of such a sign may differ from the next instance of the same sign depending on how the sign is directed or placed in the space. For instance, the verb TELL begins with the index finger in contact with the chin. If the finger moves outward toward the addressee, the verb expresses the meaning "tell

you". If it moves outward toward any female present in the room, it expresses the meaning "tell her" [Lid03].

It is useful to think of a *sign* as analogous to the concept of *word* from spoken languages [SLM06]. Signs consist of still smaller elements, a finite set of discrete meaningless elements that combined together form the signs. This discovery that a sign language has a phonemic level of structure was firstly described by Stokoe [Sto60], where a sign is divided into three *aspects*: *location*, *what acts* and *movement*.

Although Stokoe's system of representation was widely used, it did not appear helpful in understanding either phonological or morphological problems, because the model did not include some types of structures needed to describe morphological problems [Lid03].

There is now a number of other proposals concerning the sequential representation of signs.

The term *non-manual signal*, in this work referred as *non-manual component*, was introduced in order to describe aspects of signing that go beyond the hand actions [Lid77], which is referred as *manual component*.

The most important parts of *manual component* are hand shape, location of the articulation in the signing space, trajectory of hand movements, palm and finger orientations, contact of hands with the body and mutual relative position of the hands. Usually, one hand is dominant and conveys more meaning than the other hand. If the signer is left-handed then the signs are reversed with respect to the right-handed signers.

The *non-manual component* is an expression of the face and upper body poses and movements. It has similar function as intonation in spoken languages, which conveys feelings, but additionally it has grammatical functions. The carriers of the non-manual component are eyes, eyebrows, mouth, cheeks and head. For example, vertical movements of the head from top to down can express agreement, horizontal movements from one side to another can express negation of the utterance performed by the manual component of a sign [Kuc05]. Most of the rules were formed within the sign languages, but there is one exception for the use of mouth [BSS01]. *Mouth patterns* can be particular realizations of spoken language words, which are usually performed in signed modes of spoken languages.

It is necessary to differentiate terms *sign* and *gesture*. Gestures are means of non-verbal interaction among people, usually accompanying verbal speech. They range from simple actions of using a hand to point at and move objects around to the more complex ones that express some feeling (fig. 3.1) [PSH97]. There are no rules for the production of gestures in contrast to sign languages, although some gestures are internationally understandable. The gestures are not limited in space where the gesture is performed, in contrast to sign languages where the sign is limited by a *signing space*, which is delimited by the top of the head, bottom part of the torso and by the width of an elbow.

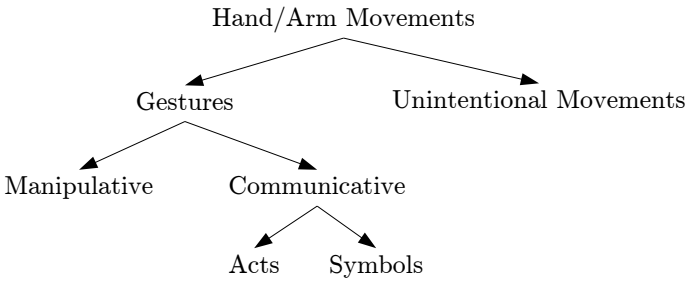


Figure 3.1: Taxonomy of hand gestures for HCI. Gestures used for manipulation of objects are separated from the gestures which possess inherent communicational character [PSH97]

Fingerspelling is part of sign languages used to represent letters of written languages (fig. 3.2). Special hand configurations denote single letters and can build whole words in a sequence. This can be used for name or abbreviation spelling.

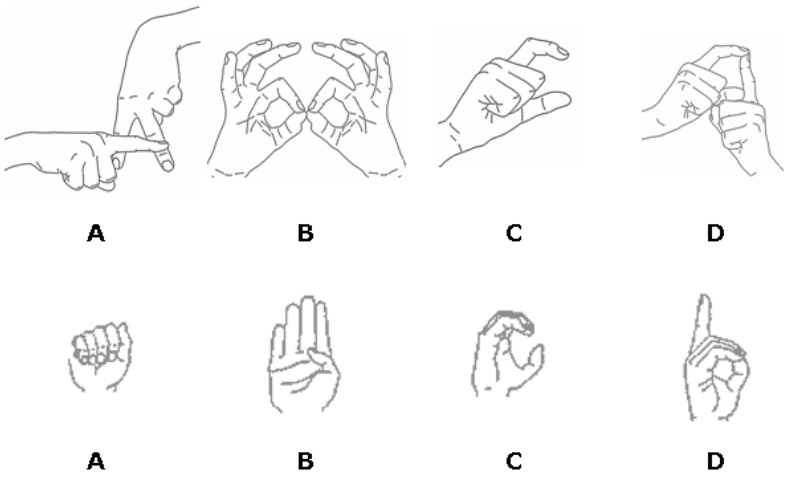


Figure 3.2: Fingerspelling - example of two-handed (top) and one-handed (bottom) alphabet

Written Forms of Sign Languages

The relation of sign languages to written form is different from spoken languages. The phonemic structure of spoken languages is sequential which leads to use of sequential phonemic writing systems. On the contrary, the sign languages have non-sequential components, i.e. many phonetic constructions are produced in the same time, which makes traditional sequential writing systems unsuitable. This is one reason why many sign languages are not written and the deaf signers use the written form of spoken language which is used in their country.

Nevertheless, several scripts for sign languages were developed. First group are phonetic notation systems, such as SignWriting (fig. 3.3) or HamNoSys (Hamburg Notational System) (fig. 3.4), which can be used for transcription of any sign language. Second group are "phonemic" systems, such as Stokoe notation (3.5), which are created for usage in a specific language.

All mentioned systems use iconic symbols. The Stokoe notation and HamNoSys are mostly restricted to linguists and academic usage, and neither of them were designed to represent facial expressions.

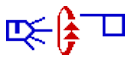


Figure 3.3: Example of SignWriting notation, Czech sign gloss CYKLUS, English sign gloss LOOP [CKK+12]

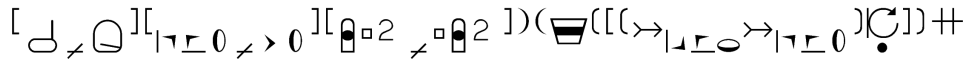


Figure 3.4: Example of HamNoSys notation, Czech sign gloss CYKLUS, English sign gloss LOOP [CKK+12]

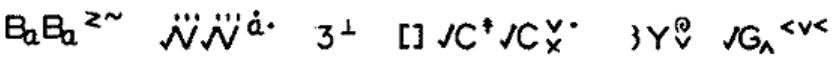


Figure 3.5: Example of Stokoe notation [wik12]

As the written forms of sign languages are immature when compared to the spoken languages, it is difficult to use these written forms in the automatic processing as there are no developed tools and the knowledge of these forms is not wide spread. Although there are some attempts of direct automatic transcriptions from a sign language utterance into a written form [EFH+12], the majority of the SLR systems uses *glosses*, which are notations of the meaning in written form of a spoken language. For example, instead of representing a sign in the written form as seen in fig. 3.3 and fig. 3.4, it is more convenient to use a gloss "CYKLUS", which is the Czech translation of the sign, or "LOOP" in English.

3.1 Sign Languages in Human-Computer Interaction

Human-Computer Interaction (HCI) is a highly interdisciplinary field studying interaction between human beings and computers or other electronic devices. The goals of the research are to find new interaction methods, make computers more user-friendly and create new automatic aids.

In the particular case of sign language, HCI is related to fields of pattern recognition, computer vision, natural language processing and linguistics. HCI can be used to:

1. facilitate communication between deaf and hearing people, by creating specialized aid devices,
2. enhance communication between deaf people and computers by using natural means of the communication, because written languages used primarily in HCI are not natural and many deaf people have poor reading and writing skills [1].

[1] For example, reading skills of 18 year old deaf students are at the level of 10 year old hearing students. [Hol93]

As the sign language is a visual form of communication, the HCI systems must be based on a processing of visual information. This has higher demands for memory storage and computational power than processing of audio information, which is used in automatic speech recognition and synthesis as another subfield of HCI, which is being developed since 1960s. Although automatic speech recognition and sign language recognition fields share many methods, the field of SLR is not so advanced. The main reason is the lack of available data and lower number of possible users. Speech HCI systems for recognition and synthesis are wide-spread and used in many applications, in contrast to HCI systems for sign language, where the field is several years behind. Current applications are able to provide good sign language synthesis (fig. 3.6) and to recognize isolated signs in videos recorded under some special conditions. Another difficulty is the necessity to translate the recognized utterances into a spoken language. This task of *automatic machine translation* is often solved concurrently with the recognition task.

The first direction of communication is from computer to the user, where sign language synthesis is used to create a 3D signing avatar which uses hand movements and facial expressing to convey utterances (fig. 3.6).

Figure 3.6: Example of sign speech synthesis [KKC+11]

The opposite direction is from the user to the computer, where some components of the sign or whole utterances are captured by an input device and recognized. In the early SLR systems, the input devices were based on contact measurements, such as data gloves (fig. 4.1). Even today this kind of device has advantage in high accuracy. The disadvantage that made these devices obsolete for real applications is cumbersome usage. This lead to the expansion of vision-based devices, such as digital cameras, camcorders or Kinect, where the measurements are not so accurate, have problems with occlusions when some body part is not visible, but are cheaper, affordable and more native. Some of the problems, such as the occlusion, can be solved by usage of multiple cameras for the cost of higher processing demands.

Sign language recognition (SLR) or its partial algorithms can be used for:

automatic translation from sign language to spoken language This is one of the ultimate tasks that incorporates continuous sign language recognition and machine translation.

search by example This task uses SLR for searching in an archive of video footages, where the search query consists of an example of a sign or a hand shape. For example, this can be used in sign language dictionaries, where a video footage with a sign or an utterance can be queried by another video footage or a single image provided by the user.

semi-automatic annotation of sign language corpora SLR systems can automatically or semi-automatically annotate sign language video footages, used for example in linguistics.

automatic collection and annotation of sign language corpora Corpora of signs can be built automatically from TV broadcasts, where the main video track is accompanied by another track with a signer who translates the original speech from audio track into sign language. Here, the correlation between the audio and sign video information is high and the SLR system could find corresponding pairs of words and signs.

education SLR systems can be incorporated into educational software, for example to create interactive tutoring systems for sign languages.

Basic schema of a continuous sign language recognition system is shown in fig. 3.7. The observed data in form of images or measurements from data gloves are recognized into a sequence of signs.

One property is that the same sign has different observations among different realizations, even of the same person. One observation of a sign can be seen as a measurement of a stochastic process, where some noise is present (fig. 3.8).

According to the model in fig. 3.9, a sign originates as a signer's mental concept, which is expressed through the manual and non-manual sign components. The observer, either a person or a device, then perceives signs as streams of visual images, which they interpret using the knowledge they possess about the sign language [PSH97].

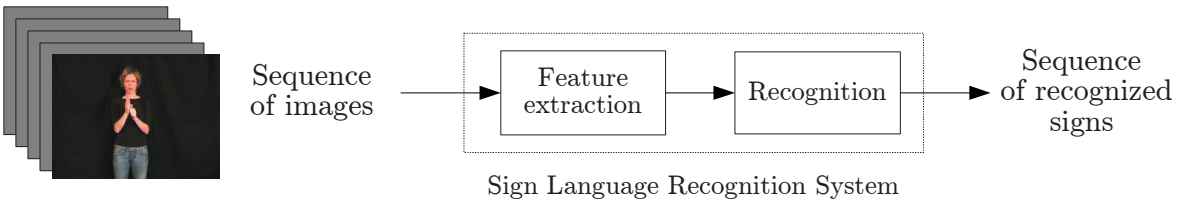


Figure 3.7: General schema of a sign language recognition system

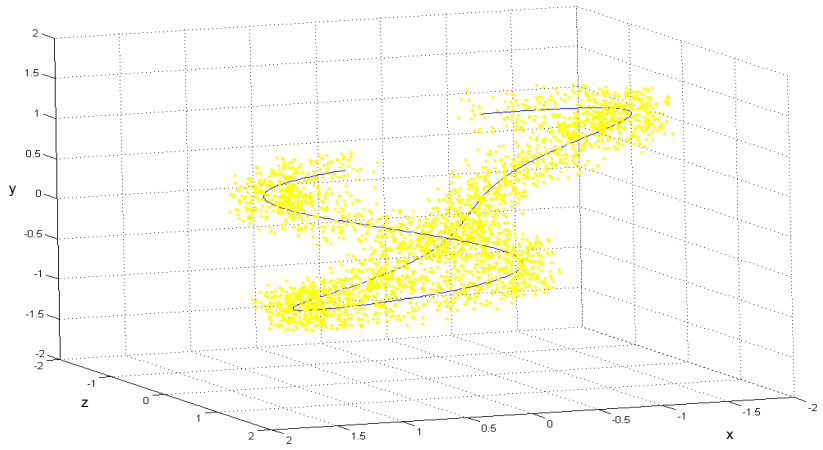


Figure 3.8: Hand movement as a stochastic process. The line represents mean trajectory of a movement, the points are measurements of several realizations of the same movement.

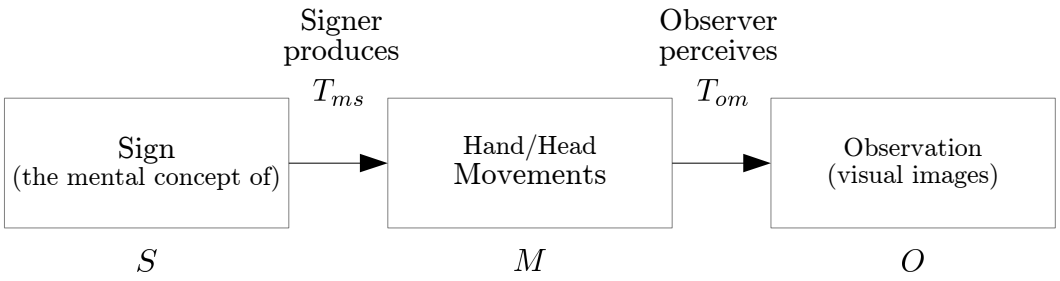


Figure 3.9: Production and perception of signs. Signs originate as a mental concept S, are expressed (T_{sm}) through head and hand motion M and are observed (T_{om}) as visual images O. Redrawn from [PSH97].

4 Data

The methods used in the field of sign language recognition are highly inspired by the ones used in the automatic speech recognition. The vast majority is based on learning of statistical models where large amount of labeled data is required. Additionally, to make the statistical models robust, the certain phenomena contained should appear many times in the data, which is not necessary true for the data collected for linguistic research. A typical set of data recorded for the purpose of linguistic research is focused on certain aspects of sign languages. It is also recorded in laboratory conditions but without any assumption that the data can be processed automatically, i.e. the illumination can be highly varying, the background is cluttered or the resolution of the footage is too low. Such conditions make tasks incorporating automatic video processing methods more difficult.

On the other hand, several sign language corpora targeted to experiments with the video processing exist, but are usually of a limited size or resolution.

Such data differ greatly from the language which is used outside of the laboratory. An example of such a real world data source is a broadcast of some public TV station, which features interpretations of some programs into sign language using an overlay box with the interpreter. This type of data source may encounter some problems with license issues. A wide variety of topics is covered in TV broadcasts and the annotation covering the whole domain is not feasible. One of the solutions is to restrict the topic to some domain where a limited vocabulary is expected, such as to weather forecasts [FSH+12].

The implicit assumption in the mentioned claims is that the data are recorded in the form of a video footage. In general, a SLR system can use any other possible type of data source. Many early SLR systems used data gloves and accelerometers for measurements of hand position and shape (fig. 4.1). Although the measurement accuracy was very high, the need of wearing

Figure 4.1: Data glove and 3D tracker used in [WCG06]

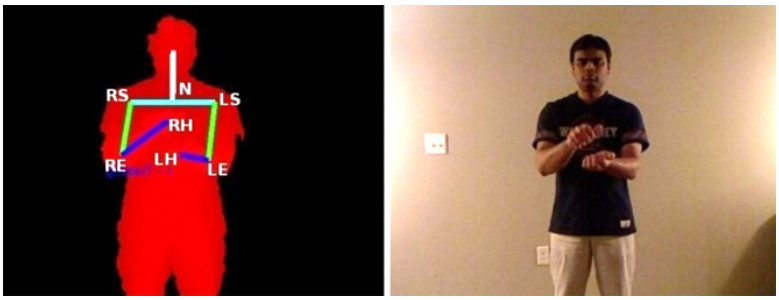


Figure 4.2: Visualization of upper body skeletal joints generated from a depth-map [ZBS+11]

cumbersome devices did not allow natural movements and thus altered the performed signs.

Therefore the use of vision sources has become more popular and is nearly exclusive data source used in SLR nowadays. The vision source uses one or multiple cameras to capture a sequence of images. In contrast to the use of data gloves, some additional problems arise: occlusion of hands and head, different appearance of the recorded signers under different lighting conditions, motion blur in case of fast movements, higher data storage and computational demands.

Optical motion capture systems can be considered as a special case of vision sources, which uses multiple image sensors to triangulate the 3D position of a signer's body parts. Such an approach was used in [KJV+12].

Most of the mentioned problems can be avoided by the usage of a depth-mapping camera, such as the Kinect [ZBS+11] (fig. 4.2). This camera provides two data streams: RGB image and a depth-map. The depth-map data allows solving majority of occlusion problems and the 3D hand pose could be determined. This kind of sensors has become reasonably priced and is a very promising direction for the use in sign language recognition field.

Apart from the type of data source, another issue should be referenced. Because the majority of the SLR systems are based on a statistical approach, a large amount of data is required. To be specific, each sign (or smaller sub-unit) should be seen several times in the data, enabling robust training of the recognition system.

To avoid the lack of training samples, a method for generating of synthetic samples can be used to enlarge the training set [JGY+09]. Another promising approach how to avoid difficult and time consuming task of manual data collection is automatic extraction of signs from existing source such as broadcasting news for the deaf. The idea is that the signs should co-occur in the similar time interval with the spoken word which corresponds to the translation of the sign. The spoken words can be either automatically recognized or extracted from manually created subtitles. Then, on a large dataset this co-occurence can be identified and some signs can be automatically extracted [SASA09] [CB09] [AAA+08].

4.1 Available Sign Language Data Sets

ASL Lexicon Video Corpus [ANS+08] (fig. 4.3a) is a public dataset containing video sequences of American Sign Language signs, with annotations of the sequences, where start/end frames of every sign is known. The vocabulary contains nearly 3000 signs, recorded from front and side view.

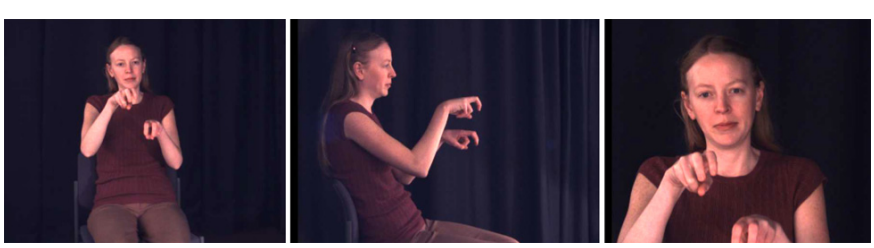
ATIS Sign Language Corpus [BSD+08] (fig. 4.3b) is based on the domain of air travel, is recorded in five different sign languages, and is suitable for recognition and translation experiments. The corpus contains 680 sentences, about 400 different signs performed by several speakers.

Auslan Corpus [Aus] consists of 300 hours of Australian sign language, with linked linguistic annotations of a small subset of the recordings.

DEGELS1 Corpus [BB12] (fig. 4.3c) is primarily targeted for linguistic research to compare annotation and analysis methods.

Dicta-Sign Corpus [EFH+12] is a recent multilingual dataset for British, French, German and Greek sign languages, providing approximately 1000 signs in every language. The video was recorded from different perspectives and with additional stereo camera. Additionally, another footage on the domain "Travel across Europe" was recorded in all four countries, by 14 to 16 informants in sessions lasting about two hours each.

ECHO Corpora [Ech] (fig. 4.3d) is a collection of three corpora in British sign language, Swedish sign language and sign language of the Netherlands. All three corpora have been linguistically annotated. For the usage in the field of sign language recognition,



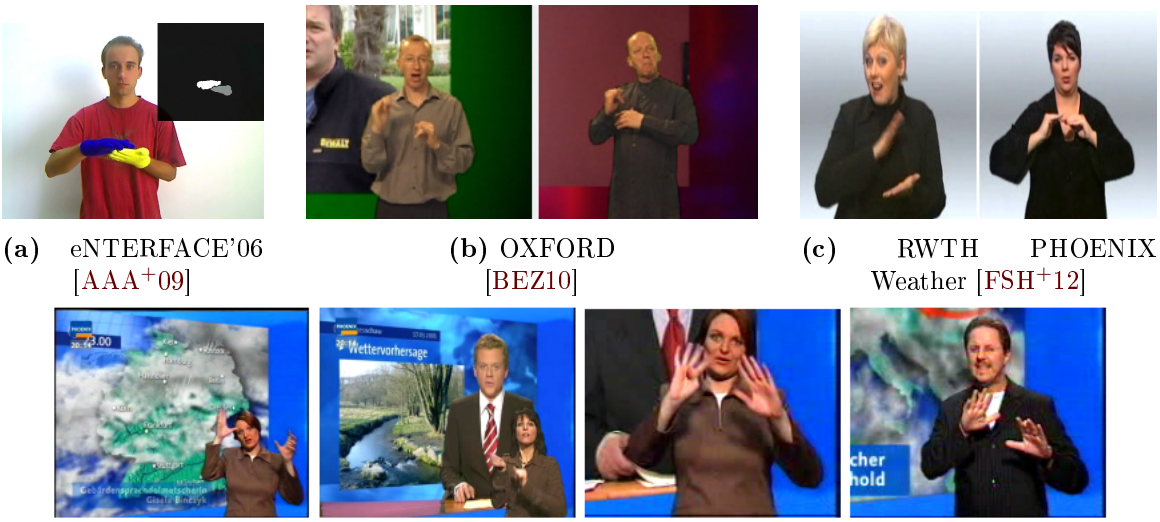
(a) ASL Lexicon [ANS+08]

(b) ATIS [BSD+08]

(c) DEGELS1 [ZDR+06]

(d) ECHO [ZDR+06]

Figure 4.3: Sample frames from different corpora



(a) eNTERFACE'06 [AAA+09]

(b) OXFORD [BEZ10]

(c) RWTH PHOENIX Weather [FSH+12]

(d) Phoenix [BSD+06]

Figure 4.4: Sample frames from different corpora

some parts of the original database were selected [ZDR+06]. Although a completely controlled environment with contrast background was used for the recording, it is hard to use the corpora for sign language recognition. The number of unique words is too high in relation to the number of utterances. Altogether, the collection contains more than 500 sentences, having 1500 unique words.

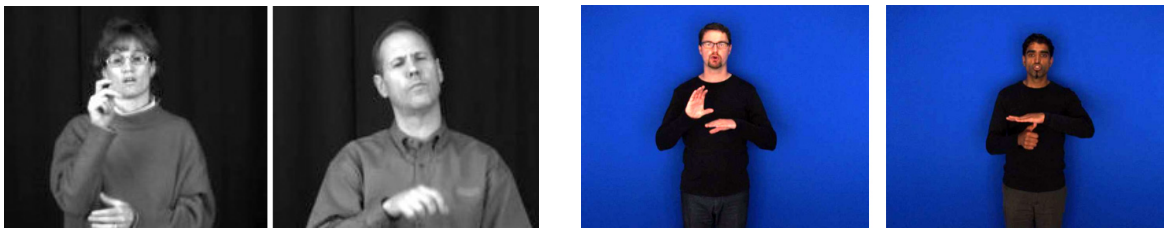
eNTERFACE'06 Corpus [AAA+09] (fig. 4.4a) is an example of a single-purpose dataset of 19 signs from American sign language, used for evaluation of non-manual effects influencing recognition accuracy. Colored gloves were used to ease the segmentation of left and right hand.

OXFORD Corpus [BEHZ08] (fig. 4.4b) is an extract from broadcast news videos recorded from BBC and is suitable mainly for tracking experiments.

Phoenix Database [BSD+06] (fig. 4.4d) is a set of 51 transcribed recordings, each is a snapshot of one weather forecast broadcast from the German TV channel Phoenix. There are 11 different signers, more than 400 sentences build from more than 600 unique words.

RWTH-PHOENIX-Weather corpus [FSH+12] (fig. 4.4c) is another corpus built from the same source of the video data as the Phoenix Database, coming from 190 weather forecast broadcasts, having 1980 sentences comprising 911 different signs from 7 signers in German Sign Language. Although not recorded under laboratory conditions, the TV studio has controlled lighting conditions, additionally signers wear dark clothes. Unfortunately, the resolution of the videos is 210x260 pixels, which might not be sufficient for some processing tasks in the field of sign language recognition.

RWTH-Boston Databases [DFN10] [DNA+08] (fig. 4.5a) is a set of several individual corpora. RWTH-Boston-50 database, containing 50 different isolated words performed by 2



(a) RWTH-Boston [ZDR+06] **(b)** SIGNUM [AK07]

Figure 4.5: Sample frames from different corpora

speakers, with about 1500 annotated frames, was used for the task of isolated sign language recognition [ZDR+06]. RWTH-Boston-104 database has been successfully used for sign language recognition [DRD+07], mainly for the evaluation of hand tracking methods. This database consists of 161 sentences, having 103 signs in the vocabulary. In more than 15k frames the signers' head and hand positions are annotated. The last is RWTH-Boston-400 database with 843 sentences, where the vocabulary size has a size of 406 words, recorded by 4 speakers. The main gloss and English translation are annotated.

SIGNUM Corpus [AK07] (fig. 4.5b) contains about 33 thousand sentences in German sign language, 700 signs and 25 different speakers. The corpus is suitable for signer independent continuous sign language recognition tasks.

The Corpus NGT [CZ08] contains 12 hours of upper body and front view recordings, having 64 thousand annotated glosses. The single sessions or pair discussions, in Dutch sign language, were recorded without a specific domain.

znaky.zcu.cz The Online web sign dictionary [CKK+12], which is primarily targeted for Czech sign language and for educational purposes, is a source of more than 3000 video files, each containing one isolated sign. This is an example of many existing online sign dictionaries targeted for educational purposes.

UWB-SLR is a set of two individual corpora, recorded in the same laboratory conditions for the usage in sign language recognition tasks. Video data were collected from three different views: front view, upper view and face view. This allows advanced experiments with automatic processing of the face and with stereo vision (fig. 4.9), the camera setup was calibrated from recorded frames containing a special calibration box with chessboard patterns.

The first corpus UWB-06-SLR-A [CHv07] (fig. 4.6) consists of 25 signs, performed by 15 signers, each sign was repeated 5 times. Several types of signs were recorded: one and two handed signs, signs containing movements with occlusions between objects, rich hand shapes and finger movements, and signs that differ only in head movement and/or face expression.

The second corpus UWB-07-SLR-P [CHT08] (fig. 4.7) contains video data of 4 signers, 378 different signs with 5 repetitions. Several types of signs are incorporated: numbers (35 signs), one and two-handed alphabet (64), town names (35), day and month names (19)

and other signs (225), which were selected as most frequent signs in the *Train timetable dialogue corpus* [KZJM06]. In total the corpus consist of 21853 video files.

Figure 4.6: Sample frames from UWB-06-SLR-A corpus (front view)

Figure 4.7: Sample frames from UWB-07-SLR-P corpus (front, face and upper view)

Figure 4.8: UWB-07-SLR-P corpus: sign SIX

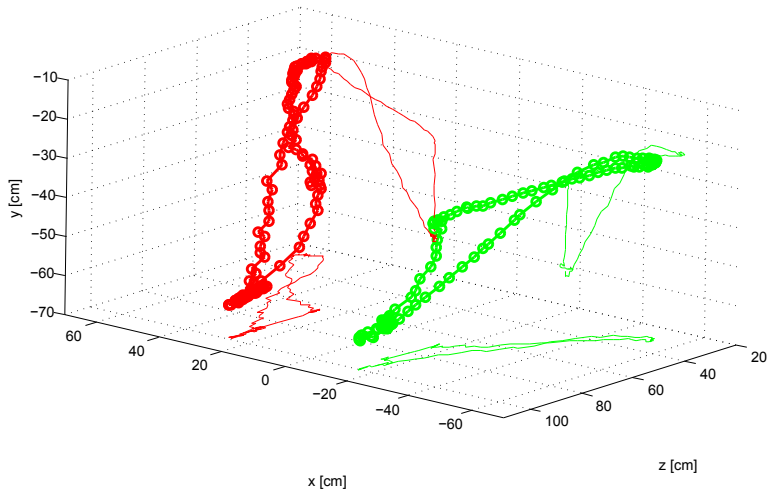


Figure 4.9: UWB-06-SLR-A corpus: English sign gloss PLANE, visualisation of 3D trajectory of hand movements estimated from front and upper camera view [CHv07]

5 Automatic Sign Language Recognition: Current Approaches and Methods

The majority of modern SLR systems is based on statistical approach, usually incorporating *Hidden Markov Models* (HMM). The scheme of such a system is shown on fig. 5.1

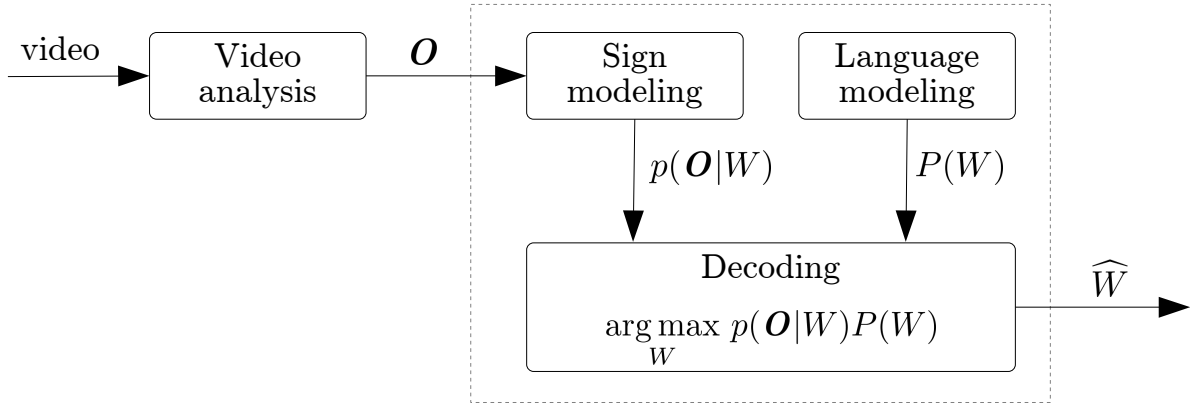


Figure 5.1: Schema of the statistical approach to automatic sign language recognition

The SLR system contains several modules. In the *Video analysis* module the video sequence is transformed into a series of observations $\boldsymbol{O} = \boldsymbol{o}_0, \boldsymbol{o}_1, \ldots$. Each observation is represented by a *feature vector*. The assumption and requirement is that the observation sequence $\boldsymbol{O}$ carries as much information needed for recognition as possible. The *Sign modeling* module converts the observation sequence $\boldsymbol{O}$ into $p(\boldsymbol{O}|W)$, i.e. into likelihoods of the observation sequence given the word (sign) sequence W. The *Language modeling* module is used to evaluate the a priori probability $P(W)$ of the given sign sequence W. Finally, the task of the *Decoding algorithm* is to find the most probable sign sequence $\widehat{W}$ for given observation sequence $\boldsymbol{O}$.

Detailed description of the particular modules follows in the next sections.

5.1 Video Analysis

The goal of video analysis is to transform the video sequence containing one signer into a series of observations $\boldsymbol{O} = \boldsymbol{o}_0, \boldsymbol{o}_1, \ldots$. Each observation is represented by a *feature vector*. To enable the use of non-vectorial image data, a feature extractor is used to map the input image data into a real valued vectorial representation.

In the next section 5.1.1 an overview of skin color segmentation techniques is presented as a first step of the processing which allows finding body parts in the video frames. Then, face detection (section 5.1.2) and tracking methods (section 5.1.3) are introduced. When the head position and hand shapes are known, the sequence of observations $\boldsymbol{O}$ can be calculated by feature descriptors applied to the hands (section 5.1.4) and the head (section 5.1.5). Optionally, a decorrelation and dimension reduction can be applied to improve quality of the feature vectors (section 5.1.6).

5.1.1 Skin Color Segmentation

For sign language recognition, the objects of our interest are head and hands. Various image segmentation methods can be used to identify pixels belonging to human body by their color. Under condition that the input image does not contain skin color objects other than human parts, image segmentation techniques can be easily used to find these objects in the image. Under another simplifying condition that the person wears clothes with non-skin color and with long sleeves, all skin color objects in the image belong to either head or hand.

For images satisfying those conditions a rule-based method proposed in [KPS03] can be used. The method uses RGB color space and a set of heuristic rules which describe the skin color cluster in the RGB color space. In addition to the original method, a parameter s was added to control strictness of the rules. With $s = 0$ the rules are the same as in [KPS03]:

$$\begin{aligned} R &> 95 + s \\ G &> 40 + s \\ B &> 20 + s \\ R &> G \\ R &> B \\ max\{R,G,B\} - min\{R,G,B\} &> 15 + s \\ |R - G| &> 15 + s \end{aligned}$$

Pixel belongs to skin color cluster if all rules are fulfilled. Example segmentation for different strictness parameter s is shown on fig. 5.2.

This rule-based method is sufficient for scenes with good and static illumination. Both corpora UWB-06-SLR-A and UWB-07-SLR-P used in this work were recorded in laboratory conditions and satisfy this condition. The quality of the segmentation was similar to the second considered method [AAA+08].

The second method uses training data with manually segmented images, where all the pixels belonging to skin areas were labeled, to train a skin color model using Gaussian Mixture Model

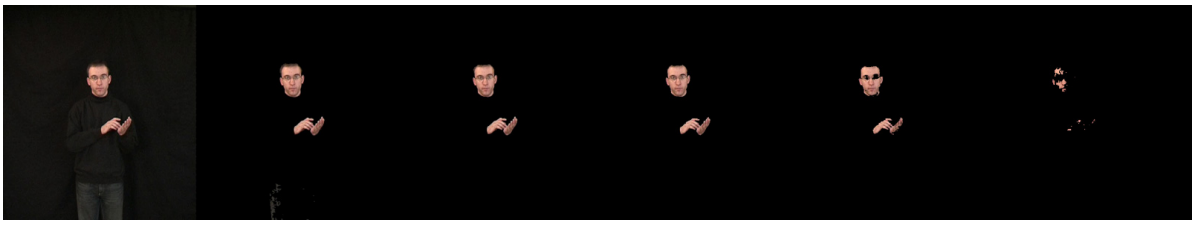


Figure 5.2: Skin color segmentation for sample image from UWB-06-SLR-A dataset. Source image (left), skin color segmentations for $s = -50$, $s = -25$, $s = 0$, $s = 25$ and $s = 50$.

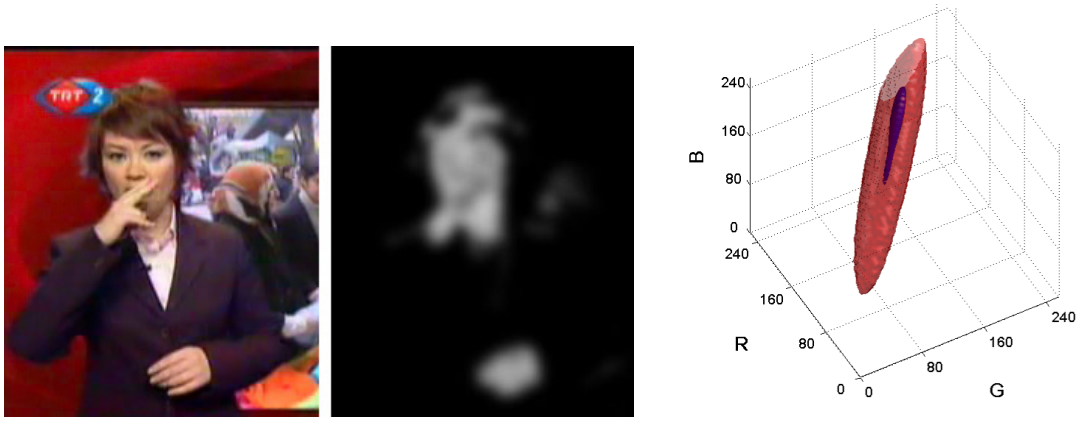


Figure 5.3: Skin color segmentation (GMM method), left: source image, middle: probability of each pixel to belong to skin color cluster, right: skin color probability distribution in the RGB color space, with two levels visible (probability of 0.5 for outside layer and 0.86 for inner layer)

(GMM). Similarly to previous method, the RGB color space is used for color representation. The training data are processed by the Expectation Maximization (EM) algorithm to train the GMM. After manual inspection of the spatial parameters of the data, five GMM components is a compromise between smooth segmentation and overfitting.

With given GMM model, the probability of belonging to a skin cluster is computed for each pixel (fig. 5.3). To segment image into two classes (skin color and non-skin color) a thresholding must be applied to the resulting probability image.

5.1.2 Face Detection

Common problem in sign language recognition is detection of face in the source image. A face detector initially proposed by Viola [VJ01], improved by [LM02] and freely available in OpenCV [ope12] software library is used in this work.

Firstly, a cascade of boosted classifiers is trained with sample views of a face (positive examples) and non-face views (negative examples). The classifier is applied to regions in an input image, with the same size as positive images. To process whole input image the search window is moved across the image and every location is classified. To find a face of different

size the scan process is repeated at different scales. The features used in classifiers are based on Haar-like features, using integral images for rapid calculation [VJ01]. As a result, a set of axis-aligned bounding boxes is computed, each containing face subimage.

5.1.3 Hand and Head Tracking

In general, *tracking* is a process of locating one or multiple moving objects over time in consecutive video frames.

Any face detection method can be used directly for *track-by-detection* face tracking. In case of some noisy detections, a running average over a series of frames is a fully sufficient and well performing way how to track a face. Particularly for sign language corpora where only one face is considered to be present in a scene. The face detection method described in the previous section 5.1.2 is partially robust to occlusions with hands. In case the occlusion prevents the detector to find the face, the largest image segment corresponding to skin-color area is expected to contain occluded face. Thus, the problem of head detection and head tracking are considered to be solved for the usage in sign language video processing.

Since the face and hands often overlap, all areas have similar color and the hand shape and its appearance is highly variable, the hand tracking is still a challenging problem.

Even state-of-the-art general tracking algorithms have some limitations, for instance they do not allow presence of multiple similar objects in the scene [KMM11], as is depicted in figure 5.4. Many specialized hand tracking algorithms were proposed to overcome difficulties with occlusions and high variability of hand shape. Mainly, methods based on similarity cost functions between the current image and a template, and methods based on dynamic hand models have become widely used [OKA12].

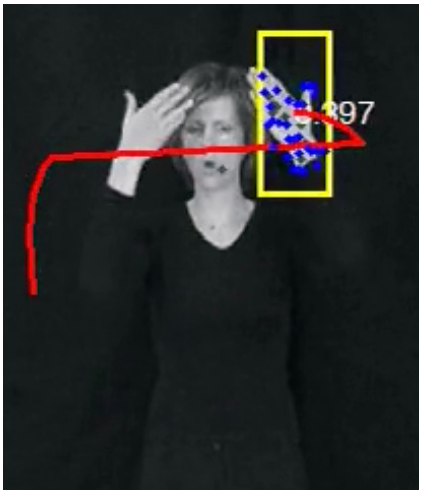


Figure 5.4: Example of tracking failure. The tracking of left hand was confused with the right hand.

For non-linear and non-gaussian problems the particle filter algorithm [IB98] has become successful. In [Ara08] the particle filter was used for hand tracking and allowed some level of robustness for occlusions, in [GC11] different hand and head models were used during occlusion and increased the tracking accuracy.

Most of the algorithms target to tracking of the center of gravity, which is sufficient for hand trajectory estimation, but is insufficient for hand shape modeling, where the exact segmentation of the pixels corresponding to hand regions is needed.

Resolving Hand over Face Occlusion

In order to describe hand appearance and shape features, it is necessary to perform hand segmentation during occlusion with face. This task is difficult because of the same color of hand and head, and large variability of the hand shape.

In [HSS02], the hand was the only object in the image and the occlusion was avoided. Methods proposed in [HLM04] [RVCB03] provide extraction of skin color regions, but they did not handle skin objects occlusions. Active contours approach [HLO05] does not cover variability and fast change of hand shape. The concept of image force field was used in [SdS07] for rough hand segmentation, which is not precise enough for sign language analysis.

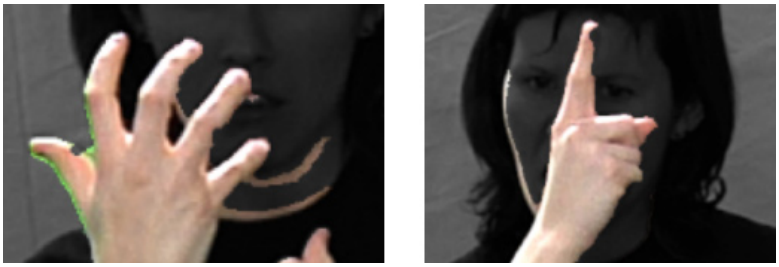
Method proposed in [GCD10] is able to segment a hand in front of a face in occlusions where the face does not change its appearance too much. In first step, immediately before the occlusion starts, the face position and the face template is remembered. Then, the face template is registered with the image during occlusion so the face position is tracked. Edges in the image are classified as belonging to the hand or to the face by mapping edges orientation to the face template (fig. 5.5 (a)). The difference between pixel colors from the template and image is measured (fig. 5.5 (b)). Robust hand segmentation is reached by merging both color difference and edge orientation difference features (fig. 5.5 (c), (d)).

In detail, the main problem is that hand segmentation in front of the face is not easily performed by only considering color feature. The idea is to find any information in the face area

(a) normalized edge orientation difference map

(b) merged (a) with mask of pixels that have changed

(c) hand segmentation result



(d) hand segmentation results with artifacts under the chin and over the collar

Figure 5.5: Hand over face segmentation [GCD10]

that was not present before the occlusion started. The assumption is that the face template, remembered from the moment just before the occlusion started, does not change its appearance during the hand occlusion. Thereby, when the face template is registered with the image, and edge filter is applied to both template and image, the orientation and presence of edges in both images on the same positions can be compared.

Several cases arise:

1. edges in the template and image are present in the same position and have the same orientation,
2. edges are in the same location but with different orientation,
3. edge from the template or image is lost.

Case 1 indicates that no hand is present at the location of examined pixel or the hand is present and has the same edge orientation. Case 2 and 3 indicates that the hand is present in the position. The method fails in case 1 where hand and face edge orientations are the same (e.g. finger parallel with face edge) and in case where the face changes its appearance in respect to the template during the occlusion (mouth moves, eye blinks, cheek movement) (fig. 5.5 (d)). This method does not limit shape or number of objects that occlude the face, thus it is suitable for sign language recognition where two hands can occlude the face at the same time.

5.1.4 Manual Component Features

As was explained in chapter 3, the manual component is one of the two sign components. In order to discriminate the signs during the recognition process, it is necessary to obtain features which describe this component, consisting of hand shape, location in the signing space, trajectory of hand movements, palm and finger orientations.

The hand tracking method described in the previous section provides necessary information about the hand location and trajectories, where hand and head positions and velocities can be used as features. The remaining hand shape and palm and finger orientations can be modeled and described by different approaches:

3D hand modeling This approach uses a 3D hand model, having several degrees of freedom, to analyze the hand posture by synthesizing the 3D model and then varying its parameters until the model and the real hand appear as the same visual images. In other words, hand model hypotheses are generated and evaluated on the available visual observations. Even though such models have become quite realistic, they can be too complex to be rendered in real-time [PSH97]. One of the first systems that can operate in real time with full articulation of two interacting hands is proposed in [OKA12]. For further SLR tasks, the model parameters, such as joint angles, can be directly used as features. Another approach for 3D hand parameters estimation uses a *motion capture system* which uses markers attached to the signer hands and other body parts, and multiple cameras for 3D tracking of those markers [KJV+12]. This system can produce accurate trajectories, but is limited for the laboratory usage only.

Appearance-based modeling Appearance based features usually do not depend on complex processing and are suitable for real-time usage. The features are extracted directly

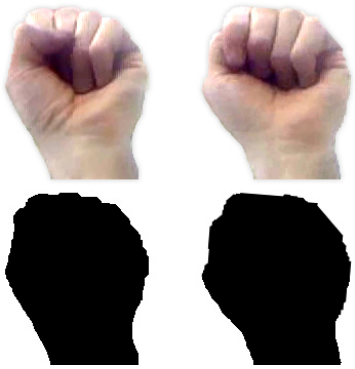


Figure 5.6: Example of different hand configurations that have nearly the same shape (represented as a 2D binary mask) from given viewpoint, but differ in the appearance

from the images. The idea is that the same hand poses have similar appearances, i.e. the captured images will be similar, which leads into similar feature vectors computed from those images. Vice versa, two different hand poses, which have different appearance in the images, should have dissimilar feature vectors. If those conditions are valid then an appearance-based method for hand pose description has discriminative power.

To describe hand shape appearance image with compact and descriptive representations of the hand configurations, both hand shape and appearance should be considered (fig. 5.6). In [RTPM10], an affine-invariant hand shape modeling is proposed using a hybrid representation of both shape and appearance of the hand.

In the following, four methods for appearance-based feature extraction are described. *Radial Distance Function* (RDF) is a shape descriptor which ignores the texture information. On the other hand, *Local Binary Patterns* (LBP) is an example of state-of-the-art texture descriptor where shape information is ignored. Despite that, both descriptors can be used and resulting features can be combined together in further processing.

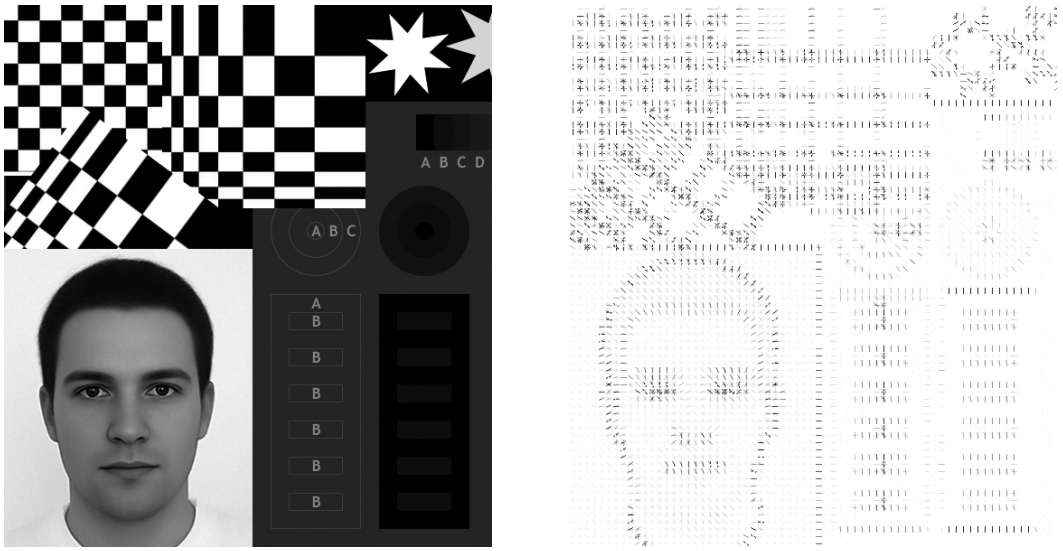
Radial Distance Function - RDF

Radial Distance Function (RDF) feature descriptor is defined as a feature vector $X_{RDF} = \{\|p_1 - p_c\|, \|p_2 - p_c\|, \ldots, \|p_N - p_c\|\}$ where $p_c \in R^2$ is the centroid of the object silhouette and p_i is the point on the silhouette contour [SB08]. $p_i - p_c$ measures the maximal extent in the particular direction denoted by i.

RDF was used in [KYSD06] for hand-based person recognition. In [KYA+11] and [HCD+11] the method was slightly modified to represent two handed shapes for the problem of finger-spelling recognition. In [SASA09] RDF was used in automatic sign segmentation problem.

Histogram of Oriented Gradients - HOG

Histogram of Oriented Gradients (HOG) is a regional feature descriptor introduced in [DT05] for object detection, in particular for pedestrian detection. The method was inspired



(a) Sample source image **(b)** Visualization of resulting HOG

Figure 5.7: Example of Histogram of Oriented Gradients - HOG

by gradient orientation histograms used in the SIFT descriptor [Low99]. In [LE09] HOG was used as a hand shape descriptor for fingerspelling recognition which outperformed other descriptors [GH06] [FR94].

The idea is to describe the local object appearance and shape by the distribution of intensity gradients (see fig. 5.7). The source image is divided into cells of the same size. For each cell, a histogram of gradient directions is computed. The concatenation of histograms from all cells forms the resulting feature descriptor.

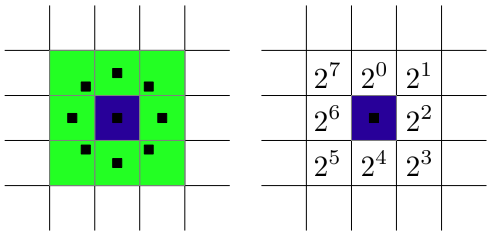
The accuracy can be improved by normalization of the contrast across a larger region of the image (called *block*), but at the cost of greater length of the feature descriptor. Because of the static illumination conditions of the corpora used in this work this contrast normalization step is not relevant and can be skipped.

The first step is the computation of the gradient values. Commonly used kernels can be used to filter image in horizontal and vertical directions: $[-1, 0, 1]$ and $[-1, 0, 1]^T$ [SHB07]. Other kernels and Gaussian smoothing could be used, but it was found that the kernels mentioned above and omission of any smoothing performed better in practice [DT05].

In the second step, histograms are created in each cell. Each pixel in the cell casts a weighted vote in a histogram bin associated to one orientation interval. The histogram bins are evenly spread over 0 to 360 degrees or over 0 to 180 degrees when the orientation of the gradient is ignored. The final feature descriptor is constructed as a concatenation of histograms from each cell.

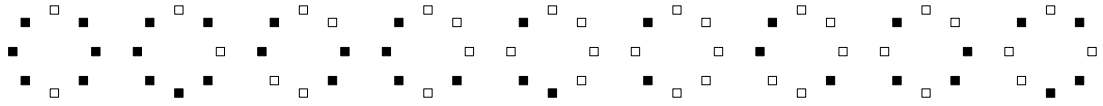
Local Binary Patterns - LBP

Local Binary Patterns (LBP) have been shown as a successful texture descriptor in many computer vision applications: face expression recognition [MB11], face recognition [CKM07],

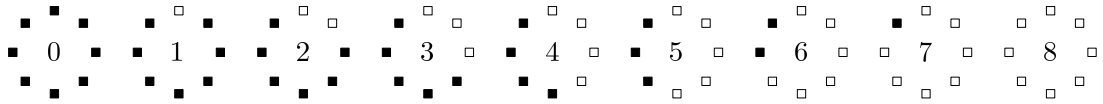


(a) 8-neighborhood **(b)** LBP mask

Figure 5.8: Local Binary Patterns - operator for 8-neighborhood with radius 1



(a) Non-uniform operator examples



(b) Uniform operators, all possible combinations without rotations

Figure 5.9: LBP - Examples of non-uniform and uniform LBP operators

human detection [WHY09] and lately in sign language recognition [HTv11] [KYA+11] [HCD+11].

Even the LBP method was firstly used for texture classification, it was shown [HCD+11] that it has a discriminative power for hand shape description. In addition, the computation is simple.

The basic form of LBP divides the source image into cells of the same size (e.g. 16x16 px). Each pixel in a cell is compared to all 8 neighbors in given order. The resulting number, which describes texture property of the examined pixel, is calculated by comparing the pixel value of the center pixel and 8 neighboring pixels, in given order. The result is a sum of 8 numbers, each set to 0 where the center pixel value is lower than of the neighbor, otherwise it is set to the mask value which is shown in figure 5.8. For each cell, a histogram of those resulting values is computed. The final feature vector is a concatenation of all histograms, i.e. the vectors representing individual histograms are concatenated into a single vector.

The method can be generalized to use different number of neighboring pixels at different distance from center pixel.

To reduce the length of the feature vector and to increase robustness in locations with multiple edges, so called *uniform LBP* were introduced as another extension to the basic method. The inspiration is that some binary patterns occur more often than others. A local binary pattern is marked as uniform if the binary pattern contains at most two transitions from 0 to 1 or vice versa (see fig. 5.9). All non-uniform patterns are labeled with the same number. Thus, the length of the feature vector is reduced from 256 to 59 (when using 8-neighborhood).

Another extension which introduces rotation invariant description is available, but not

considered here since the hand orientation is important feature and rotation invariancy is not desired.

High-level Linguistic Feature Descriptor

A high level description of hand shape and motion was introduced in [BWK+04] and later used in [CB09] [CB10].

When hand trajectories are known, its coordinate representation can be converted into a phoneme representation taken from sign linguistics [BWK+04]:

HA Position of the hands relative to each other

TAB Position of hands relative to key body locations

SIG Relative movement of the hands

DEZ Shape of the hands

This notation provides a high-level binary feature descriptor. The events like *hands move apart*, *left hand on right shoulder* are examples of events covered by the descriptor.

In this work, the **DEZ** subgroup of the features was ignored and only positional features were used (see tab. 5.1).

HA	TAB	SIG
1. Left hand high	6+7. Face	20+21. Hand makes no movement
2. Right hand high	8+9. Left side of face	22+23. Hand moves up
3. Hands side by side	10+11. Right side of face	24+25. Hand moves down
4. Hands in contact	12+13. Left shoulder	26+27. Hand moves left
5. Hands crossed	14+15. Right shoulder	28+29. Hand moves right
	16+17. Chest	30. Hands move apart
	18+19. Stomach	31. Hands move together

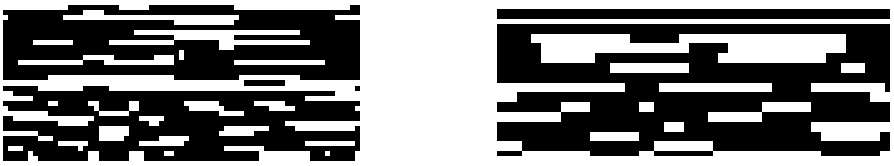
Table 5.1: High level linguistic features [BWK+04]. The features in pairs denote two standalone features for left and right hand.

The positions of the hands used for feature computation are calculated from the centroids.

In contrast to the original method, where only features related to the dominant hand were used, both hands are considered and included in the features in this work. Examples of such feature vectors calculated for a set of images of one sign is shown in fig. 5.10.

5.1.5 Non-manual Component Features

As was explained in chapter 3, the non-manual component is important part of the sign. To allow robust recognition and to distinguish signs which have the same manual component and differ only in the non-manual component, it is necessary to obtain features which describe this



(a) sign AUTOBUS **(b)** sign CESKE BUDEJOVICE

Figure 5.10: Examples of high-level linguistic feature descriptor for a short video sequence. Each column corresponds to one video frame, row to particular feature (white color denotes active feature).

component and are able to discriminate non-manual expressions, such as raised or lowered eyebrows, head tilts, eye gaze, etc. In particular, eyebrows indicate negation and question schemes, mouth and lip movements encode adjectives and adverbials, and head pose has participation in affirmations, questions and denials.

A key factor for estimation of non-manual signals is accurate tracking of the facial landmarks [MLY+12]. Although many models have been proposed for face tracking, most of them target global shape optimization and are sensitive to occlusions of the face (fig. 5.12).

Common methods used for face tracking are Active Shape Models (ASM) [CTCG95] and Active Appearance Models (AAM) [ETC98]. Other methods use a 3D deformable model [vAKK08], but this method cannot be applied in real-time systems.

In the following, *Active Appearance Model* method employed in this work is described.

Active Appearance Model (AAM)

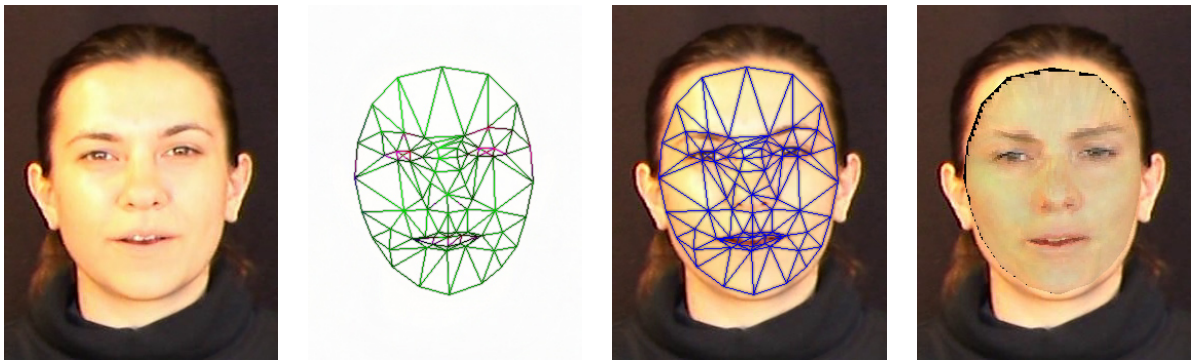
Active Appearance Model (AAM), firstly proposed in [ETC98], is a non-linear generative parametric model, most frequently applied for face modeling [LTC97].

The goal of the algorithm is to match a statistical model of object *shape* and *appearance* to a source image. The model is built in a training phase from a set of images with annotated coordinates of landmarks. It is related to the *active shape model* (ASM) [CTCG95] that uses only shape information and thus does not take advantage of all the information available.

The first step is to fit the AAM to the source image, the model parameters are found to maximize the match between the model instance and the input image [MB04]. The model parameters can be used directly as feature vector, representing the face shape and appearance, or can serve as a basis for calculation of higher level features, like *lip height*, *left eyebrow elevation* etc.

Fitting an AAM to the source image is a non-linear optimization problem [MB04]. For the experiments in this work, an efficient implementation based on *Inverse Compositional Image Alignment* [BM01] [BM04] is used.

Figure 5.11 presents an example of an AAM fitting. The AAM in the example was trained on different people than the one on which the AAM was applied, thus the appearance (fig. 5.11d) modeled by the fitted AAM seems to belong to other person. When no other objects occlude with the head, the AAM fits the face correctly, but the fitting can fail when larger part of the face is occluded (fig. 5.12).



(a) Source image, UWB-06-SLR-A corpus

(b) AAM shape model

(c) Fitted AAM *shape* projected to source image

(d) Fitted AAM *appearance* projected to source image

Figure 5.11: Active appearance models (AAM): example of a model and fit of the model to the source image

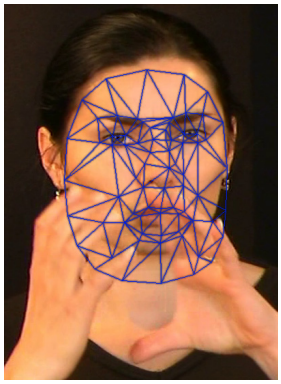


Figure 5.12: Example of AAM wrong fitting due to occlusion

5.1.6 Feature Decorrelation and Dimension Reduction

Features obtained by the feature extraction methods described in the previous sections are likely to be correlated, i.e. the features statistically depend on each other. In such a case the sign modeling module (section 5.2) has to process some redundant information, which is time and memory inefficient, and full covariance matrix of features should be used for Gaussian Mixture Model (GMM) used in sign modeling (section 5.2.1).

Dimensionality reduction is a way how to deal with both problems. It aims at selecting the most discriminative information from the feature vectors, usually by use of a linear transformation of the feature space. Although this transformation leads to a loss of information, the parameter estimation in the reduced feature space is often more reliable.

In the following, Principal Component Analysis (PCA) method employed in this work is explained. The first use is to reduce and decorrelate features obtained from appearance-based features described in previous sections. The second use is its application for *Eigensign* features obtained directly from image pixel values (section 5.1.6).

Comparison of other methods can be found in [THZ+08]. The results in [Zah07] show that PCA and its Eigensign application is a powerful transformation method and is more useful to improve the resulting error rate when compared with *Linear discriminant analysis* (LDA) method.

Principal Component Analysis

The Principal Component Analysis [Pea01] (PCA) is widely used method to turn a set of correlated variables into a smaller set of uncorrelated variables. The high-dimensional dataset described by correlated variables can be reduced to only a few meaningful dimensions, which account for most of the information. This method finds the directions with the greatest variance in the source data, called *principal components*.

Denote a set of random vectors with observations as $X = \{x_1, x_2, \ldots, x_n\}\}$, $x_i \in R^d$. The mean value vector μ and covariance matrix are computed from all observations as:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad , \quad S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \tag{5.1}$$

The eigenvalues λ_i and eigenvectors v_i of matrix S are computed to satisfy: $Sv_i = \lambda_i v_i$, $i = 1, 2, \ldots, n$.

The resulting eigenvectors are ordered by their eigenvalue in descending order. The k principal components are selected as the eigenvectors which correspond to the k largest eigenvalues, denoted as

$$P = (v_1, v_2, \ldots, v_k) \tag{5.2}$$

The problem of determining k is not trivial, either relevant information is lost (for low k) or noise is included (for high k).The problem of determining k is widely discussed in [PNJS05].

PCA is applied to an observed vector x to obtain k-dimensional projected vector y:

$$y = P^T(x - \mu) \tag{5.3}$$

Original data can be reconstructed from y by a backprojection from k-dimensional into original d-dimensional space:

$$\hat{x} = Py + \mu \tag{5.4}$$

Eigensigns

For some tasks, image itself can be used directly as a feature descriptor (only by reshaping the image matrix into the feature vector). The problem with such an image representation is its very high dimensionality. If we consider each pixel as one dimension, then, for instance, an image with 100x100 pixels lies in a 10000 dimensional space. Such a dimension is too high to be used in further processing. Since not all dimensions are equal and some carry more information about the performed sign, we are looking for those dimensions that account for

most of the information. The technique *eigensigns* applied here for images with a signer is analogous to *eigenfaces* which is widely used in face recognition problem for encoding face images into lower dimensions. The only difference is the form of the data, for sign language recognition the images contain a signing person in contrast to *eigenfaces* where images with faces are used. The main idea is to use PCA method (section 5.1.6) to describe the images with a signing person by only a few variables that account for most of the information. The idea was developed by Sirovich and Kirby 1987 [SK87] and used for face recognition by Turk and Pentland 1991 [TP91], and was used for sign language recognition by Dreuw [Dre12].

The method is sensitive to variation in lighting and scale. The images, containing a signing person, must be normalized in position and size. Any face detector can be used to find face positions and normalize the image so that the face is a center of new coordinate system and the size of the face is used to normalize the image size. To avoid problems with lighting variation the source data should be recorded in the static lighting conditions.

Denote set of training images as $\mathbf{I} = \{I_1, I_2, \ldots, I_n\}$, each of size h x w. The images contain signing person in normalized position and size.

Images are reshaped from two dimensions into vectors X, each with dimension $d = h \times w$, by concatenating all image columns into one vector.

After PCA is applied to X (eq. 5.2) and transformation matrix P is known, any image with the same dimension can be projected into the PCA subspace (eq. 5.3). This representation of the original image has much lower dimension and is suitable for further processing.

In practice, the size of covariance matrix S (eq. 5.1) is huge even for low dimensional images (e.g. S has size 10000 x 10000 for images of size 100 x 100 pixels). Additionally, number of images in the training set I is often smaller than dimension d. Thus the matrix X of size $d \times n$, where $d > n$, can only have $n - 1$ non-zero eigenvalues. In this case it is possible to use the eigenvalue decomposition $\overline{S} = X^T$ of size $N \times N$:

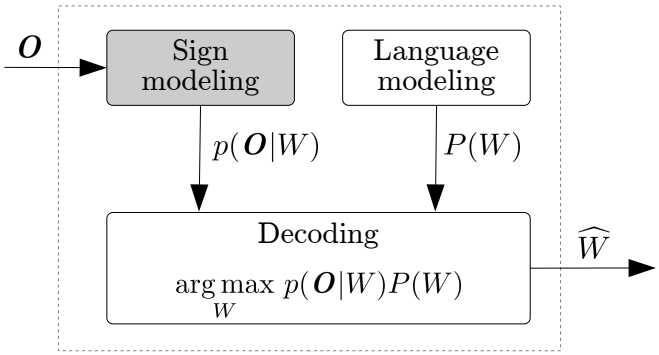
$$X^T X v_i = \lambda_i v_i \tag{5.5}$$

The original eigenvectors of S are computed with a left multiplication of the data matrix X:

$$X X^T X v_i = \lambda_i X v_i \tag{5.6}$$

The resulting eigenvectors are orthogonal, normalization is needed to get orthonormal eigenvectors [DHS01].

5.2 Sign Modeling



The *Sign Modeling* module is used to calculate an estimation of the likelihood $p(W|\boldsymbol{O})$ for given sequence of observations $\boldsymbol{O}$ and sequence of signs W.

Researchers used methods such as *Finite State Machines* (FSM) [HTH00], *Artificial Neural Networks* (ANN) [Vam96] or *Dynamic Time Warping* (DTW) [HRH07]. The most popular tool is *Hidden Markov Models* (HMM) which is widely used in automatic speech recognition for acoustic modeling since the middle of 1970s [Bak75] [JBM75]. The *acoustic modeling* is based on an idea that the vocal tract state is stationary in a short time interval called *microsegment*. A short acoustic signal is produced and can be represented by a set of features [PMMR06]. Analogously, the same concept of short stationary segments is applied in sign language recognition, where the state of head and hands is expected to be stationary if the microsegment is short enough.

The HMM is generally used for modeling of a stochastic process that in discrete time intervals generates two time-aligned sequences of random variables. The first is a sequence of states of the Markov model between which the process transitioned, the second sequence is a sequence of observations $\boldsymbol{O} = \boldsymbol{o}_0, \boldsymbol{o}_1, \ldots$. The sequence of states corresponds directly to the recognized sequence of signs. The number of recognized signs and their alignment in time is contained in the result. With this property the HMMs overcome other methods like FSM and ANN where another method must be used for determination of sign bounds, which is not a trivial task.

The sign language recognition based on HMM can be formalized as a task to determine the most probable sequence of states, between which the HMM transitioned, given the observation sequence $\boldsymbol{O}$. The task of best estimation of $p(W|\boldsymbol{O})$ consist of selection of HMM topology and of selection of observation probability function type and its parameters.

For the task of time series modeling, where the hidden state evolves through time, the left-right (Bakis) topology is used almost exclusively (fig. 5.13).

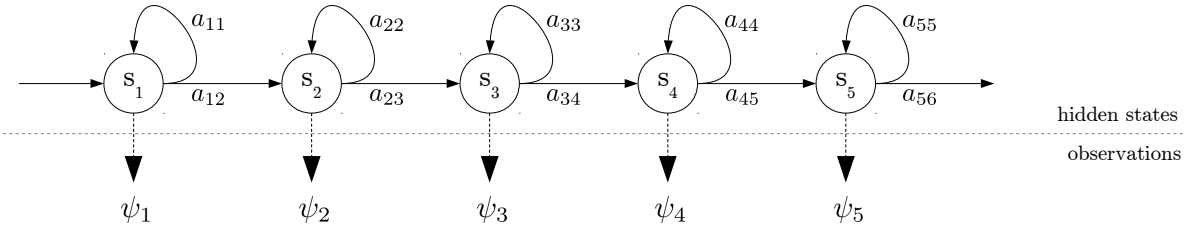


Figure 5.13: Example of a 5-state HMM of a sign in left-right (Bakis) topology

For given observation vector $p(W|\boldsymbol{O})$ the HMM starts in the first state and either stays in the same state or makes a transition into the state with higher index. For the last observation the HMM ends in the last state. Thus, in left-right topology, the state cannot go back in time.

For the case of small vocabulary, i.e. small number of signs that can be recognized, one model is created for each sign. Generally the number of states can be different for each model, however, a fixed number of states can be used, with comparable results [RS83].

sub-units The previous approach becomes unfeasible in case of larger vocabularies with more than a hundreds of signs. Instead of using one model for each sign, smaller sub-sign units are used. The inspiration came again from automatic speech recognition where syllables, phonemes or even smaller units are used. The most widely used sub-word unit is *triphone*, a special case of a context-dependent phoneme unit, which is able to model inter-word and intra-word coarticulation context.

The sign models are created by linking the sub-sign units, each unit can be used in multiple signs. This provides a way how to increase the vocabulary size and deal with more realistic conditions where the data are collected. In [VM99], the signs were broken into basic phonemes using the idea of the Movement-Hold model. The absence of general linguistic understanding of sign phonetics prevents the use of manually designed sub-units, similarly to manually designed triphones which are used in speech processing. Thus the data-driven approach for modeling the sign sub-units is widely used. In [FGGC04], the signs were segmented into sub-units by K-means clustering applied to trained sequences of HMM states. Another idea for data-driven creation of sub-units was proposed in [HAS09] with the assumption that the hand movement always goes through three phases: deceleration, acceleration and uniform motion.

variation of articulation Two sources of variations appear in the sign language utterances. The first is due to signer variability where two signers articulate the same sign differently. In order to solve this issue: a) signer independent models must be employed, b) already trained signer dependent models must be adapted, c) the models must be trained from the whole set of possible signers. Second source of variation is due to co-articulation, where the realization of the current sign is influenced by the previous and following signs. The solution of this issue is: a) to learn all the possible variations of a sign from the data, b) to consider each variation as a different sign.

HMM parameters When the topology of the HMM is known, and either whole sign or sub-unit modeling was chosen, two sets of HMM parameters must be determined. The first set, *transition probabilities*, consists of elements a_{ij}, each defining the probability of transition from state i to state j. The common representation of transition probabilities is a matrix $\boldsymbol{A} = |a_{ij}|$, where a_{ij} is zero for the pairs of states which are not linked, thus for left-right topology the matrix $\boldsymbol{A}$ is sparse.

The second set of parameters is known as *observation probability* functions. The observation probability function associated with i-th state is commonly represented as a continuous probability density function $\psi_i(\boldsymbol{o}|\boldsymbol{\lambda})$, where $\boldsymbol{o}$ is an observation vector and $\boldsymbol{\lambda}$ is a vector representing the parameters of the probability distribution function, which are estimated during the training stage. The type of the function is selected a priori. In the field of automatic speech recognition two types of the function are commonly used: Neural Network Densities Functions and Gaussian Mixture Models [Trm12]. The latter type is the most used in the field of SLR.

5.2.1 Gaussian Mixture Model (GMM)

In statistics, a general mixture model is a probabilistic model where the underlying data belong to a mixture distribution, whose density function is a linear combination of other probability density functions:

$$p(x) = \sum_{i=1}^{M} w_i p_i(x) \tag{5.7}$$

Each individual density function $p_i(x)$ is called *mixture component* (or *Gaussian* for normal distributions), and the weight w_i associated to i-th density function is called the *mixture weight*, all weights sum to one:

$$\sum_{i=1}^{M} w_i = 1 \tag{5.8}$$

The most common mixture model is the *Gaussian mixture model* in which the mixture components are Gaussian (normal) distributions, each having its own mean and variance parameters.

The observation probability function ψ used in HMM can be modeled as a multivariate normal distribution by a Gaussian Mixture Model:

$$\psi(\boldsymbol{o}|\boldsymbol{\lambda}) = \sum_{i=1}^{M} w_i p_i(\boldsymbol{o}|\boldsymbol{\lambda_i}) \tag{5.9}$$

where $p_i(), i = 1, \ldots, M$ is a mixture component, $\boldsymbol{\lambda_i}$ is a subset of $\boldsymbol{\lambda}$ associated to i-th mixture component and w_i is the weight of the i-th component. Given an N-dimensional observation vector $\boldsymbol{o}$, each mixture component is modeled as an N-dimensional normal probability distribution:

$$p_i(\boldsymbol{o}) = \frac{1}{(2\pi)^{\frac{N}{2}}\sqrt{det\ \boldsymbol{C}_i}}\ exp\left(-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_i)^T\boldsymbol{C}_i^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_i)\right) \tag{5.10}$$

where $\boldsymbol{C}_i$ is a covariance matrix of size $N \times N$, μ_i is a mean values vector of length N.

With GMM, the subset $\boldsymbol{\lambda_i}$ of $\boldsymbol{\lambda}$ parameters associated to i-th mixture component is defined as $\boldsymbol{\lambda_i} = (w_i, \boldsymbol{\mu}_i, \boldsymbol{C}_i)$.

The optimal number of mixture components M must be determined experimentally. In real world applications some simplifications are made when using GMMs, such as the $\boldsymbol{C}_i$ matrices are assumed to be diagonal. This can reduce the amount of data needed in the training stage and speed up the calculation.

5.2.2 Details on HMM Training and Recognition

Commonly used Baum-Welch algorithm [RJ86] is used to **train** the HMM parameters. The algorithm uses an iterative expectation-maximization (EM) algorithm to find a HMM which is a local maximum in its likelihood to have generated a set of training observation sequences. The training is a batch process where example observations of a given sign are used as inputs to this algorithm.

In the **recognition** process a set of trained HMM models (for each sign or sub-unit) is used to decode the input observation sequence $\boldsymbol{O}$ via the Viterbi algorithm. The result consists of a sequence of most probable signs W, or sub-units, together with their respective starting and ending positions. In case of isolated sign language recognition, only a subset of W containing one sign is considered. Beside the original Viterbi algoritm, a functionally equivalent *Token Passing* algorithm [YRT89] [YEG+06] is widely used.

5.2.3 Fusion Strategies

To increase the quality of a recognition system, multiple information sources can be combined together. To handle multiple information streams that are present in sign language, like left and right hand movements and shapes, and head movements and expression, independent modeling of each stream can be employed.

The sign modeling, as described in previous sections, implicitly uses the given sequence of observations $\boldsymbol{O}$ in the form of one feature vector for each video frame. The feature vector is usually concatenated from three particular feature subvectors belonging to two hands and head, and each subvector can be another concatenation originating from different feature descriptors. This type of fusion, performed at feature level, is called *early fusion* or *feature fusion* [AHEK10]. For example, hands positions, RDF and LBP hand feature descriptors and AAM-based features can be all merged into a single feature vector, which represents the signer state in the examined microsegment. This fusion approach is advantageous that it can utilize the correlation between multiple features from different modalities and requires only one learning phase.

However, the time asynchrony between the streams is hard to represent. The opposite type of fusion, performed at decision level, is called *late fusion*. Several recognition subsystems provide resulting decisions based on individual feature subsets. Those decisions are then combined and analyzed to obtain a final decision.

Both approaches can be combined. For example, in [ABCA09], a sequential belief-based fusion was proposed, at first using a decision of a HMM with fused manual and non-manual features, and in case of a hesitation another HMM with only non-manual features was used to make the final decision in a cluster of similar signs.

There are various extensions to HMMs that explicitly model several processes occurring in parallel, thus staying in between the early and late fusion: *middle fusion*. In [VM99], *parallel HMM* (PaHMM) is presented, which models the parallel processes independently in a way that they can be trained independently too. PaHMM were shown to be more scalable than previously used *factorial HMMs* (FHMMs) [GJ97] or *coupled HMMs* (CHMMs) [BOP97], which require training examples of every conceivable combination that can occur in parallel. The most recent work [TKM09] brought *product HMM* into SLR (fig. 5.14), which has been

previously successfully applied in audiovisual speech recognition, allowing two streams to be in asynchrony within the model but forces them to be in synchrony at the model boundaries. It also permits to control the degree of asynchrony between the streams. This particular work presented promising results for SLR.

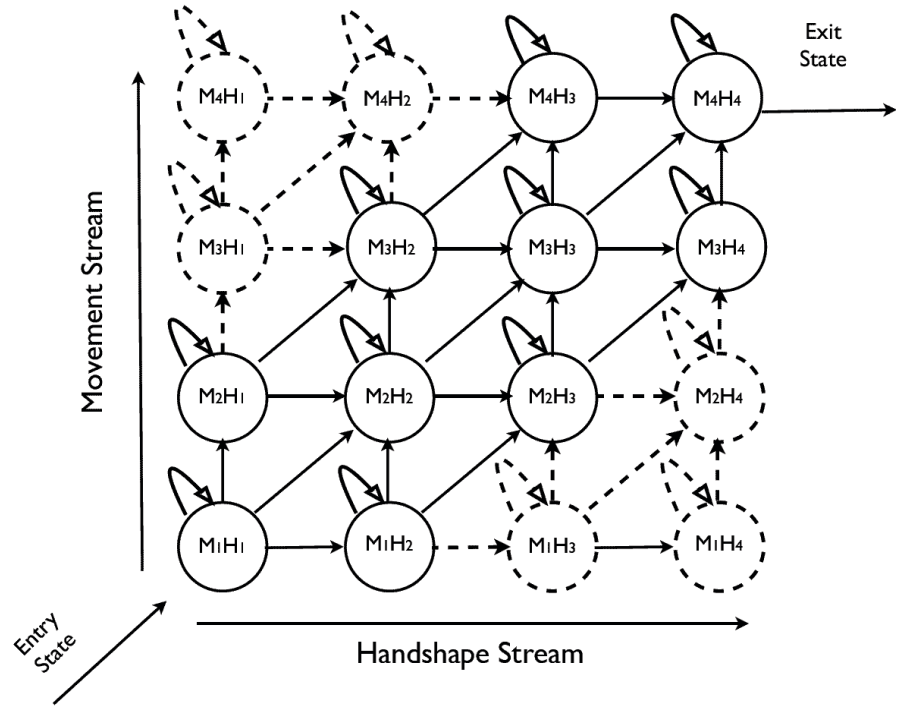
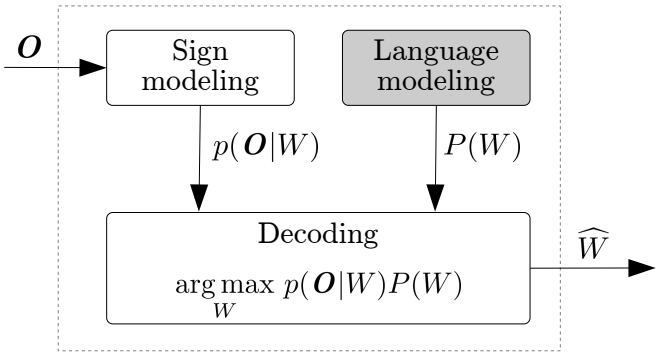


Figure 5.14: Example of a product HMM with 2 streams and 4 states in each stream. The movement and handshape streams are denoted by M_x and H_y, where x, y are the states of the movement and handshape stream model respectively. [TKM09]

5.3 Language Modeling

The task of the *language modeling module* is to compute an estimate of the a priori probability $P(W)$ for any given sequence of signs W. This probability $P(W)$ is used in the next stage by the decoding algorithm.



Due to a limited amount of training data available, this thesis is primarily focused on sign modeling problems. Thereby only a brief overview is given.

In the case of isolated sign language recognition, the most common is uniform language model which assigns the same probability $P(W)$ to each sign from the vocabulary. Thus the whole recognition process is based on the sign modeling only.

In the second case of continuous sign language recognition a broad range of the language model types can be used [GMW97]:

uniform language model The same probability is assigned to each sign. This implies the probability of any sentence W having exactly N signs: $P(W) = (1/V)^N$, where V is the number of signs in the vocabulary.

finite state language model The set of legal sign sentences W is represented as a finite state network (or regular grammar). Each path through the network generates a legal sign sequence.

n-gram language models All sign sequences are possible. The probability of the predicted sign depends only on the $n-1$ immediate predecessor signs. A special case when $n = 0$ is called *zerogram language model* and is equal to the uniform language model described above.

All the language model types are technically ready to be incorporated into sign language recognition systems. However, there is a lack of available training data. This is in contrast to the field of automatic speech recognition, where, for instance, the language model can be build from the data automatically mined from the web [vHSV11].

5.4 Sign Decoding Methods

The *decoding algorithm*, depicted as a module in fig. 5.1, finds the most probable sequence of signs $\widehat{W}$ for given observation sequence $\boldsymbol{O}$:

$$\widehat{W} = \arg\max_{W} p(\boldsymbol{O}|W)P(W) \tag{5.11}$$

The first factor $p(\boldsymbol{O}|W)$ is provided by *Sign modeling* module, the second $P(W)$ by *Language modeling* module.

For high number of possible sequences W the direct and exhaustive optimization of the eq. 5.11 is impossible. In such a case only a limited fraction of most promising hypotheses can be evaluated.

As the probability densities provided by the sign and the language models are not necessarily commensurable, the equation 5.11 is modified in the real world applications:

$$\begin{aligned} \widehat{W} &= \arg\max_{W} p(\boldsymbol{O}|W)\ P(W)^{\beta}\ \gamma^{L} \\ &= \arg\max_{W} log\ p(\boldsymbol{O}|W) + \beta\ logP(W) + L\ log\gamma \end{aligned} \tag{5.12}$$

where the parameter β is called *language model weight*, γ is *sign insertion penalty* and L is the number of signs in the current hypothesis W.

The γ and β parameters are estimated experimentally. The language model weight β is used to balance the influences of the language and sign models. The sign insertion penalty γ is used for adjusting the ratio between the number of the insertion and the deletion errors (see next section 5.4.1).

5.4.1 Accuracy Evaluation

The quality of a recognition system is mostly evaluated by the Levenshtein distance [Lev66], which is defined as a minimum number of edit operations needed to transform the recognized sequence of signs $\widehat{W}$ into the reference utterance W_{ref}, which contains N signs. The edit operations allowed are *insertion*, *deletion* and *substitution*. Denoting the number of insertions I, the number of deletions D, the number of substitutions S and the length of W_{ref}, two widely used measures are defined: correctness *corr* and accuracy *acc*:

$$corr = \frac{number\ of\ correct\ signs}{N} = \frac{N - D - S}{N} \cdot 100\% \tag{5.13}$$

$$acc = \frac{number\ of\ correct\ signs - I}{N} = \frac{N - D - S - I}{N} \cdot 100\% \tag{5.14}$$

Instead of the *acc* measure, the *word error rate* WER can be used:

$$WER = 100 - acc = \frac{number\ of\ edit\ operations}{N} = \frac{D + S + I}{N} \cdot 100\% \tag{5.15}$$

reference:	KDY	LOUCIT_SE	CHUTNAT	HORSI	LOUCIT_SE	HORSI	DIVKA	AUTO		PLZEN
recognized:		KOLENO	PLZEN	HORSI		NEJHORSI	DIVKA	AUTO	KOLENO	PLZEN
error type:	D	S	S	-	D	S	-	-	I	-

$N = 9$, $D = 2$, $S = 3$, I=1 $\Rightarrow$ **corr** = 44.4%, **acc** = 33.3%, **WER** = 66.7%

Table 5.2: Example: *correctness* and *accuracy* calculated on a synthetic sentence

Example of a *acc*, *corr* and WER calculation on a random synthetic sentence is shown in tab. 5.2.

One common problem in the field of sign language recognition is the difficulty to compare results with different authors, mainly due to use of different corpora and absence of an agreed benchmarking corpora, although there are some attempts for hand tracking evaluations [DFN10].

5.4.2 Confidence Intervals

The accuracy *acc* is an important measurement of system quality, but it does not make any statements about the stability of the performance. For example, we could be interested in an interval of accuracy in which the recognition system provides 95% of the results over the time. *Confidence intervals* (CI) are tools used in statistics that allows to estimate probability that the observed value will fall in a confidence interval.

There are many methods for CI estimation. For a result set with normal distribution $\mathcal{N}(\mu, \sigma^2)$, about 68.3% of the results lies in the interval $(\mu-\sigma, \mu+\sigma)$, 95.5% in $(\mu-2\sigma, \mu+2\sigma)$ and 99.7% in $(\mu-3\sigma, \mu+3\sigma)$. This is known as *three-sigma rule*, or *empirical rule*.

However, this approach is not suitable for the recognition accuracies, where the distribution is non-gaussian. In such a case or when the distribution is unknown, a *bootstrap* method proposed in [BN04] can be used. The method is intuitive, precise, easy to use and makes no assumption about the distribution of errors. It can be applied to find confidence intervals on word error rate WER in speech recognition evaluations.

The core idea of the *bootstrap* method is to create replications of a statistic by random sampling from the dataset with replacement (so-called Monte Carlo estimates) [BN04]. The original dataset (of recognition results) is divided into s segments and sampled randomly B times, typically $B = 10^3 \ldots 10^4$. The point estimate is computed from these samples. Finally, B point estimates are used to determine the confidence interval, for example the value $P_{-0.05}$ which is the 2.5 percentile and $P_{0.05}$ which is the 97.5 percentile construct the 95% confidence interval $(P_{-0.05}, P_{0.05})$. Despite the simplicity, the method works well in most cases.

5.5 Search by Example

The aim of a *search by example* system, alternatively *sign look-up* system, is to allow a user to perform a sign or a component of it, and to search in a sign language database on the basis of features extracted from the user performance [EFH+12].

Although there are many sign language dictionaries using video to present sign lemmas [1], most of them allow to search lemmas based on either groupings according to basic hand shape or typing the gloss in the case of bilingual dictionaries. This makes the search task difficult for the dictionaries having a large number of lemmas.

In the *Czech Sign Language online dictionary* (`http://signs.zcu.cz`) [CHL+10], SignWriting or HamNoSys symbols can be used for the search. Despite the usefulness, the drawbacks are that the search works only for manually annotated lemmas with SignWriting or HamNoSys notation. The second is that many of the users are not familiar with those notation systems.

Considering that most of the sign language dictionaries contain videos with a signer performing a single sign or some longer utterances, sign language recognition methods can be employed for the searching purposes.

In [ANS+10], an application that lets the user submit a video of a sign as a query, and presents the most similar signs from the manually annotated database, was proposed, based only on hand centroids and dynamic time warping. In [WSM+10], a similar tool is proposed, with the search based on features extracted from hand motion and hand appearance. The similarity between signs was measured by a combination of dynamic time warping and a similarity measure based on hand appearance. In user-independent experiment, with a vocabulary of 1113 signs, the correct sign was included in the top 10 matches for 78% of the test queries.

In [ECG+11], a *Search-by-Example* proof-of-concept prototype is presented that uses an interactive sign recognition system based on depth images from Kinect device to perform search in four language corpora, with sign dependent recognition rates above 70% on 984 signs.

[1] A *Lemma* is the dictionary form of a word.

6 Automatic Sign Language Recognition: Proposed Approach and Results

This chapter reveals contributions and results of this work, particularly in video analysis and sign modeling. Multiple approaches for the sign language recognition task are introduced and their performances are compared. Then, proposed the *search by example* system is introduced and evaluated.

6.1 Data

All the experiments were performed on two own corpora described in section 4.1:

UWB-06-SLR-A [CHv07] 25 different isolated signs, recorded by 15 signers, each sign repeated 5 times.

UWB-07-SLR-P [CHT08] 378 different isolated signs, recorded by 4 signers, each sign repeated 5 times

After the manual verification of the corpora, some flawed videos were removed from the sets, such as videos where the signer was smiling or where the direction of sight was wrong.

Only the frontal views were used, which results in the same recording conditions as are used in sign language dictionaries, which are partly targeted by this work. Thus, all investigated methods should be applicable for other recordings and corpora that contain one signer recorded from the front view.

In many works, the datasets are split into training and testing subsets, and used in all experiments invariably. Here, the datasets are provided as a whole and are split on demand, for example by a cross validation approach.

The processing of both corpora resulted in some intermediate datasets that can be used elsewhere:

UWB-06-SLR-A hand corpus is an image set of 74009 pairs of left and right hand images (see examples in fig. 6.1), extracted by the tracking process from videos in UWB-06-SLR-A corpus, with manual verification.

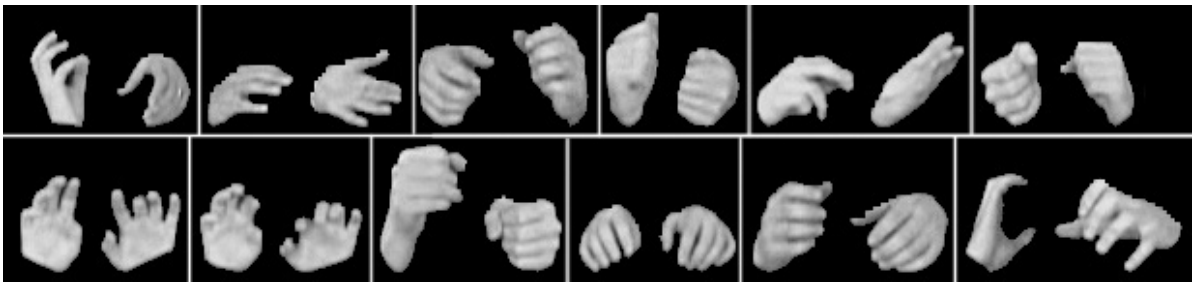


Figure 6.1: UWB-06-SLR-A hand corpus: examples of hand pair images

UWB-07-SLR-P hand corpus has the same properties, but contains 128677 pairs of images from UWB-07-SLR-P corpus.

6.1.1 Image Normalization

The videos in datasets were normalized before further processing. Face detector, described in section 5.1.2, was applied for each video, resulting in a set of bounding boxes that enclose image part containing a face. Since the videos contain always only one signer, maximally one bounding box is expected in each frame. From all the bounding boxes in all frames from one video a mean bounding box was calculated, corresponding to the average position of the face.

All videos were normalized so that the mean face position established an origin of new coordinate system and the width of the mean face bounding box determined new scale. After the normalization, all sign realizations are performed in the same coordinate system (see example in fig. 6.9).

6.2 Feature Extraction

In this section, several approaches for feature extraction from videos containing sign language utterances are described. As was theoretically described in *Video Analysis* section 5.1, the goal is to transform the video sequence into a series of observations $\boldsymbol{O} = \boldsymbol{o}_0, \boldsymbol{o}_1, \ldots$, where each observation is represented by a *feature vector* $\boldsymbol{o}_i$.

The *Eigensigns* method (section 6.2.1) uses no high-level knowledge about the content of images and the features are directly based on pixel values. This approach was used in several works (see section 5.1.6).

Before employing higher level features, it is necessary to perform head and hands tracking, which is described in section 6.2.2. This tracking information is employed in calculation of manual component features extracted from hands (see section 6.2.3), and in calculation of non-manual features extracted from face (section 6.2.4).

6.2.1 Eigensigns

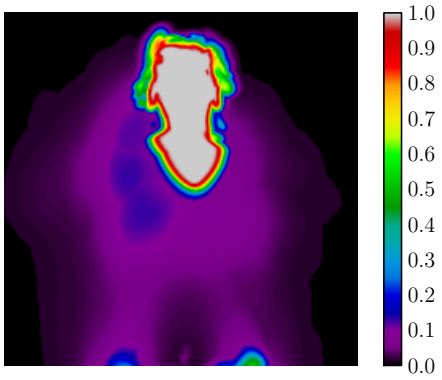
The *eigensigns* method was used as a baseline method for feature extraction. The method is based on appearance-based approach described in section 5.1.6. Since the UWB-06-SLR-

A and UWB-07-SLR-P corpora used for experiments were recorded under invariable lighting contitions, this method does not fail due to illumination variance. Another requirement of the method is to use position normalized images (as described in section 6.1.1).

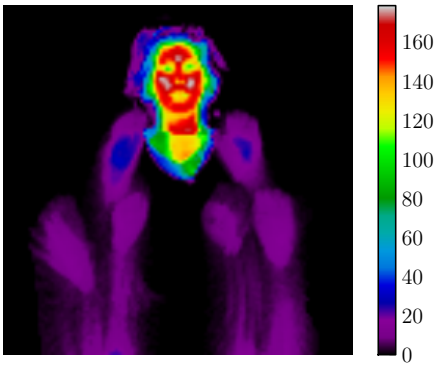
In the experiments, the images were normalized to fixed sizes: 60x60, 80x80 and 100x100 px. To optimize calculation of transformation matrix P (eq. 5.2), only pixels with non-zero prior probability to contain skin color values are considered (fig. 6.2a). In UWB-07-SLR-P, only 59% of pixels contain at least one skin color value through all the frames in all the videos, the remaining 41% of pixels can by ignored. This greatly lowers the dimension of covariance matrix S.

The mean shape and resulting principal components (eigensigns) can be seen in figures 6.2b and 6.2c.

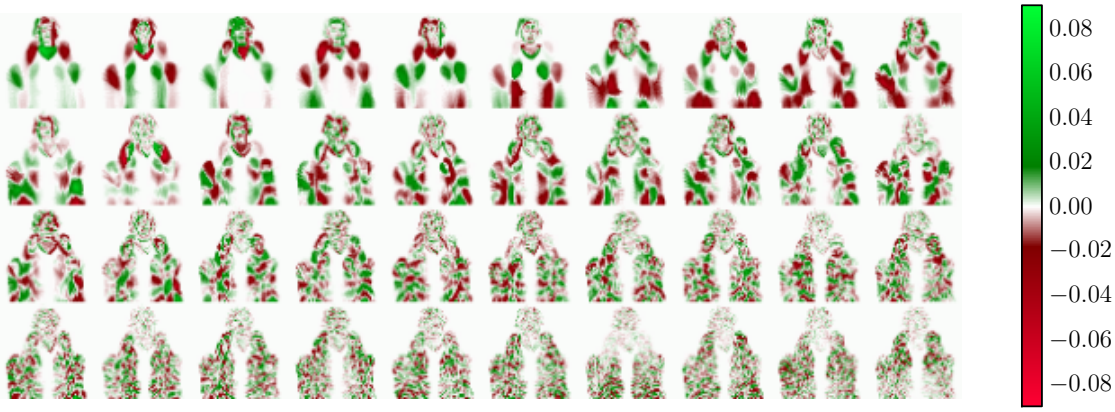
Now, any image can be represented as a linear combination of k eigensigns, which span a k-dimensional subspace of the original image space by choosing a subset of eigenvectors. This resulting subspace, so-called *sign space* (analogous to *face space* used in eigenface method), has



(a) Signing space: prior probability of each pixel to contain skin color value



(b) PCA: mean



(c) PCA: visualization of selected eigensigns (1..10, 11..30, 30..70, 70..150)

Figure 6.2: Visualization of eigensigns method applied on a subset of UWB-07-SLR-P dataset

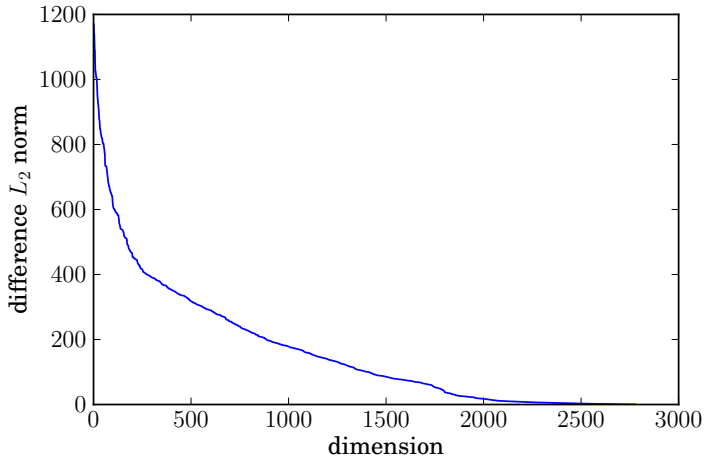


Figure 6.3: Absolute difference L_2 norm of original image and PCA backprojected image in dependence of k principal components (*dimension*).

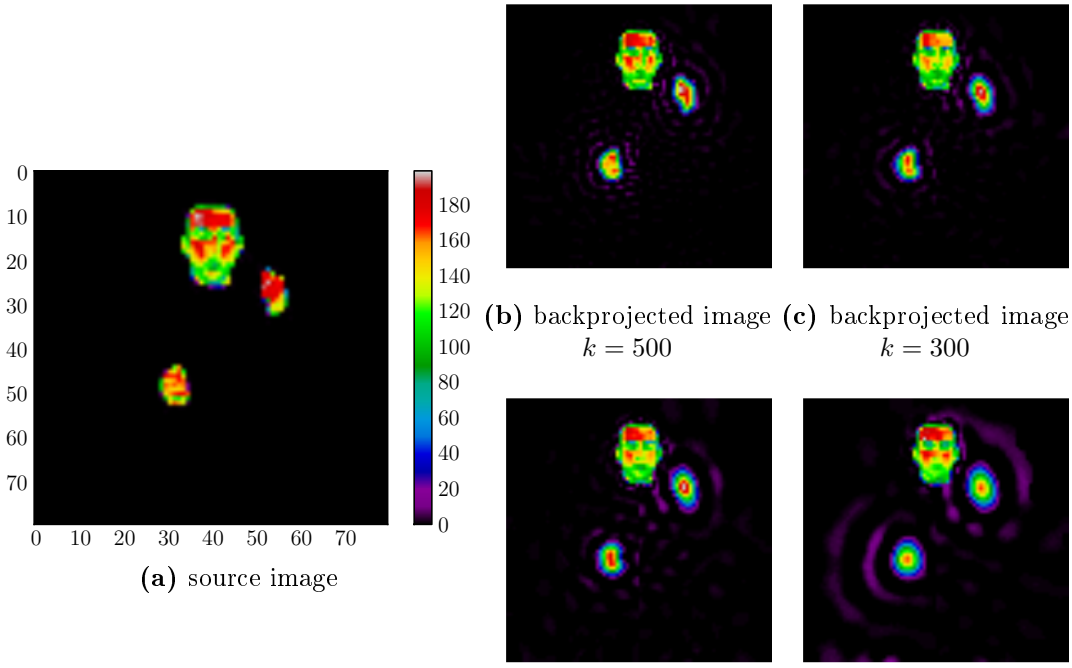
its origin in the mean sign and the axes are the eigensigns.

The problem of determining k [PNJS05] was solved experimentally. For few randomly selected images, the absolute difference L_2 norm was calculated for source image and image backprojected from PCA projection, where k principal components were used. This L_2 norm corresponds to reconstruction error, i.e. it explains how much image information is lost depending on selected k. The dependence of the norm on k is shown in fig. 6.3 as mean results for 100 random images from UWB-06-SLR-A dataset, image size 60x60 px, where PCA was computed on 10000 images. The graph shows that selecting k larger than 2000 brings no much better backprojected image and that for $k < 300$ the error grows rapidly. The results in the graph suggest to use k around 300 as a compomise between low dimensional representation of the image and reconstruction error.

Visualization of backprojected images for different k is shown on fig. 6.4. It confirms that for $k < 300$ much information is lost, mainly in hand areas, which are crucial for successful sign recognition.

The *sign space* represents well images similar to those on which the principal components were computed. Any other content can be represented in this *sign space* too, but the projection into this subspace does not represent these images well. This can be seen in fig. 6.5, where synthetic images containing a circle in different positions are projected to signing space and backprojected to original image space and visualized. It is evident and expected that the linear combination of eigensigns, used to represent the original synthetic image, does not work well for these images of different content. Similar situation with erroneous representation would happen for images with signing person performing signs in different locations and poses that were not present in the image set from which the eigensigns were computed. Thus, for reliable representation of all signs from whole dataset it is crucial to compute eigensigns from large number of images.

The *Eigensigns* method used as a baseline method for appearance-based feature extraction is only one example of possible similar methods using orthogonal linear transformation that



(a) source image

(b) backprojected image $k = 500$

(c) backprojected image $k = 300$

(d) backprojected image $k = 200$

(e) backprojected image $k = 100$

Figure 6.4: PCA: visualization of backprojected images for different k. The grayscale images are visualized using a color map for a better insight.

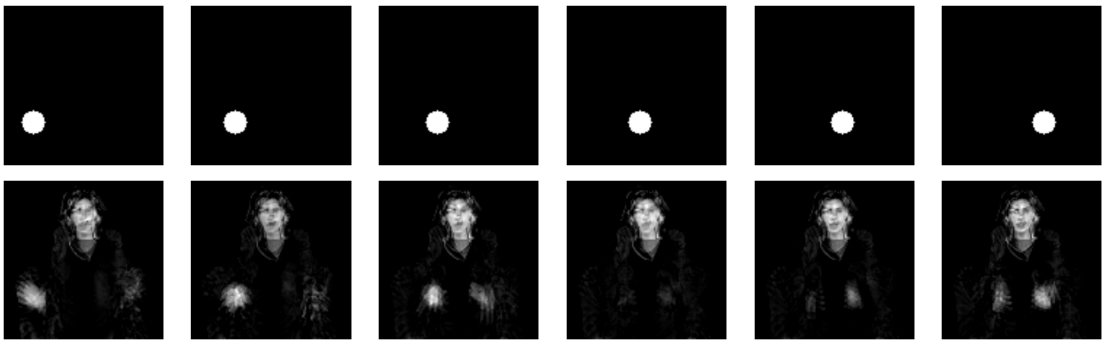


Figure 6.5: Top: source synthetic image, bottom: PCA backprojected image.

transforms the image space into a new space. Another method available is Linear Discriminant Analysis (LDA), which transforms original image space in such way that the discrimination between classes is maximized [Fod02]. Experiments performed in [Dre12] showed that PCA typically outperforms LDA based feature space dimension reduction in the field of sign language recognition. All these methods use low level appearance-based approach to extract features from the source image without any other knowledge about the content.

6.2.2 Head and Hands Tracking

To allow feature extraction of higher level hand and head features, it is required to segment parts of image belonging to those body parts. A novel approach that separates head and hands segments, and combination of tracking and hand classification was employed. The approach produces two results:

hand tracking and segmentation All hand areas in the image are segmented, tracked and identified whether the area belongs to left or right hand, or to an area with occlusion of both hands.

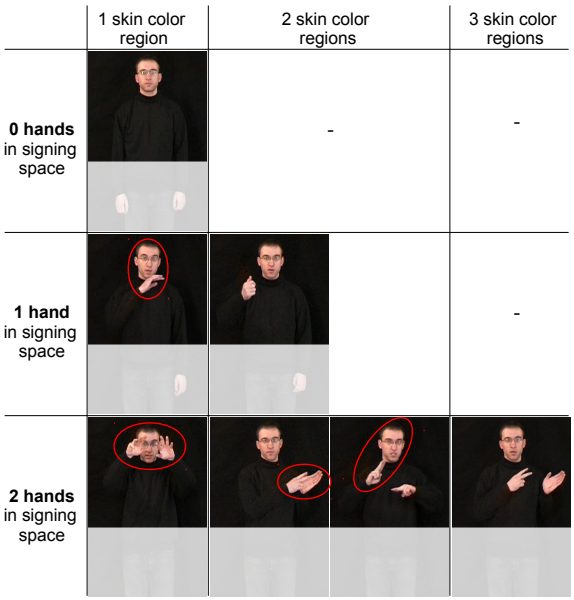
head segmentation and occlusion removal Head area is segmented and in a case of occlusion with hands, the head area is approximated by best matching template, so that the resulting head image is never covered by an occlusion.

Figure 6.6: Example of head and hands separation process. Top: original frame, middle: resulting separated image with head (is approximated during occlusion), bottom: resulting separated image with hands.

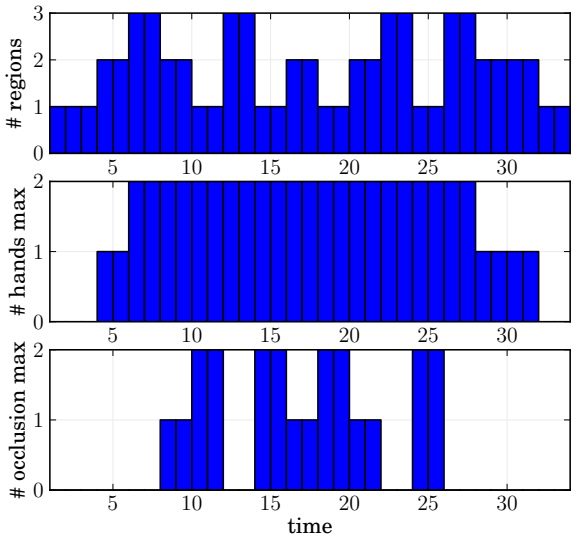
An example is shown in fig. 6.6, where occlusion of both hands and the head is present. The detailed description of the algorithm follows.

Occlusion Detection

Before the head and hands separation step, the following rule-based method is used to mark frames with possible occlusion. It is needed for the collection of face templates. The method determines the numbers of coherent skin-color regions (*blobs*) that are present in the signing space. With the prior knowledge about the dataset, it is known that the video contains exactly one blob corresponding to the face. No prior assumption is posed about the hands that can be even outside of the video frame. A rectangular area, called *signing space*, is placed in manually designated position with respect to the face mean position. All subsequent processing is done only in the signing space and the rest of the image content is ignored. In terms of image processing, the signing space is a *region of interest* (ROI) .



(a) Number of visible skin color regions depending on number of hands in the signing space. The bright area under the main body denotes a space that stands out of the signing space. Red ellipses highlight occlusions.



(b) Example of varying number of skin color regions (blobs) appearing in the signing space over time: **(top)** number of skin color regions, **(middle)** maximal potential number of hands located in the signing space, **(bottom)** number of potential occlusions

Figure 6.7: Occlusions of hands and head

The figure 6.7a depicts all possible combinations of hand presence in the signing area and number of visible blobs (skin color regions). When the number of hands plus one face equals the number of blobs, no occlusion is present in the frame (see the cases in fig. 6.7a on the diagonal). An occlusion occurs in the other cases when the number of visible blobs is lower.

The identification of the first and last frame of the occlusion is performed as simple comparison between number of blobs and expected number of hands that are present in the signing space, as depicted in fig. 6.7a. The figure 6.7b is an example of this process, where the top graph displays number of visible blobs B in the signing space during the time. The middle figure shows maximal number of hands that can be present in the scene, the values are based on the analysis of number of skin color regions in time, and are calculated as $min(max(B_{past}), max(B_{future})) - 1$, where B_{past} denotes vector with number of blobs in the previous frames and B_{future} in the following frames. For a two-handed sign, the values can be maximally 2, for one-handed maximally 1.

The final result, which identifies frames standing before and after possible occlusion, as seen in the bottom figure 6.7b, where the number of potential occlusions is computed from the values in the middle and top graph:

$$number\ of\ potential\ occlusions\ in\ frame = max\ number\ of\ hands - (number\ of\ blobs - 1)$$

Frames, where the number of potential occlusions is greater than zero, determine frames with any kind of occlusion (hand+hand or head+hand) or frames where a hand left temporarily the signing space. In these frames the following separation algorithm is employed, using face templates from the frames just before and after this occlusion.

Head and Hands Separation

The method described in section 5.1.3 [GCD10], that resolves hand over face occlusion, is used to segment hands in the image. In contrast to the original method, template matching method is used to match face template and the source image. Because the position of the signer is normalized in the source video, the searching window is limited to the area where the head can occur. In extension to the original method another template of the head after the end of the occlusion is used. This partly reduces constraint of this method that the head appearance should not change much during the occlusion. Both templates are used in further processing and the one with the highest template matching score is used. The head template images are rotated in several angles to allow head rotation in the image plane.

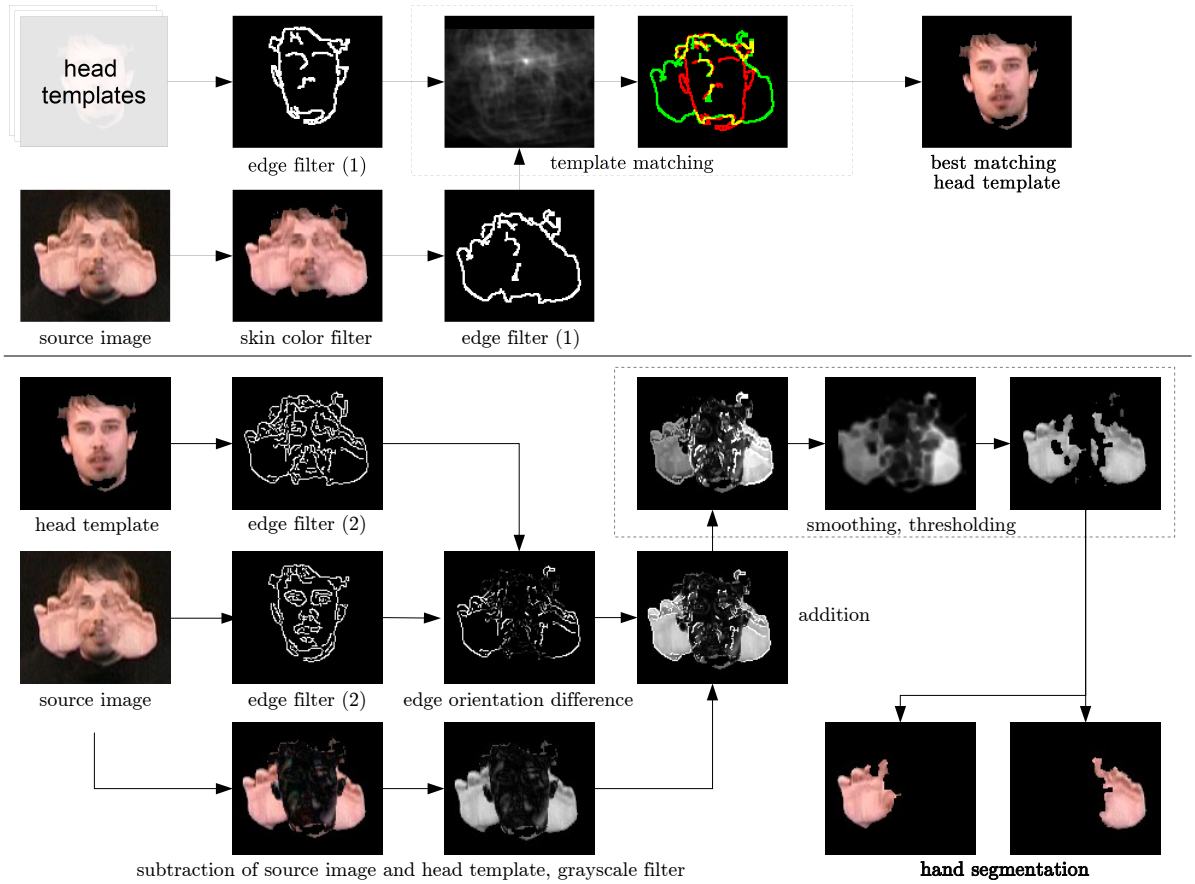


Figure 6.8: Occlusion resolving schema. Top: head template matching. Bottom: hand segmentation.

The whole proposed hand and face segmentation process is depicted in fig. 6.8.

If the number of skin color areas decreased between two adjacent frames, then a hand either left the signing area or occluded with another hand or with the face. In all cases this is considered as first frame where a possible occlusion started, even when the hand left the signing area. In such a case the result of the hand segmentation is an empty frame.

In the opposite case when the number of skin color areas is increased, either a hand entered the signing area or left an occlusion with another object.

The novel idea is to separate any source frame into two independent image channels, first containing the head and the second the hands. In non-occlusion image, the separation is straightforward because the head position is known and hands are visible as standalone blobs. If occlusion is present, the proposed separation process is performed. The best matching template is used as an approximation of the appearance of the occluded face and is used as the first output channel. The hand segmentation is used to find the pixels corresponding to the hands and are used as the second output channel.

Each of the two channels can be used in later stages of sign language analysis. It allows using more trivial methods for hand tracking that expect no occlusion with head. Likewise, methods for head expression analysis, that fail when the head is occluded, can be used seamlessly. This is a huge advantage but for the price that some information from the face expression can be lost by approximation of the head appearance by the template during the occlusion.

Hands Tracking

The hand tracking is performed in video frames where the face area was removed by the previously described algorithm and thus the frames contain only hand areas. When a hand enters the signing space, which is discovered by detection of new skin color region, new object tracking is established with simple rules. In the subsequent frame, the nearest skin color region to the previous position is associated with the tracked object. In case of collision, i.e. when multiple objects are associated with the same object, the tracking of all objects is suspended and objects in the current frame establish new tracking.

The result of this simple tracking is a set of tracks, each following a movement of one skin color area, which can consist of one hand or an occlusion of two hands. For example, in fig. 6.10 there are 5 tracks, where two hands moved up, then occluded, separated again and moved down. Now, the only missing information of each track is the association to either left or right hand. Hand classification is employed to associate the tracks to particular hands, and is

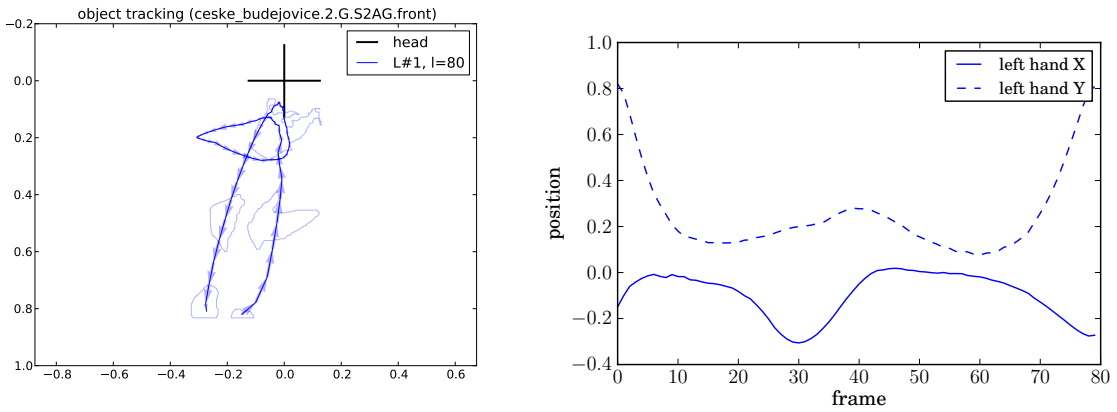


Figure 6.9: Examples of hand tracking, one-handed sign CESKE BUDEJOVICE. Black cross is the origin of the coordinate system placed in the center position of the head. In the left figure, the trajectory is shown together with few hand contours randomly selected in the time sequence.

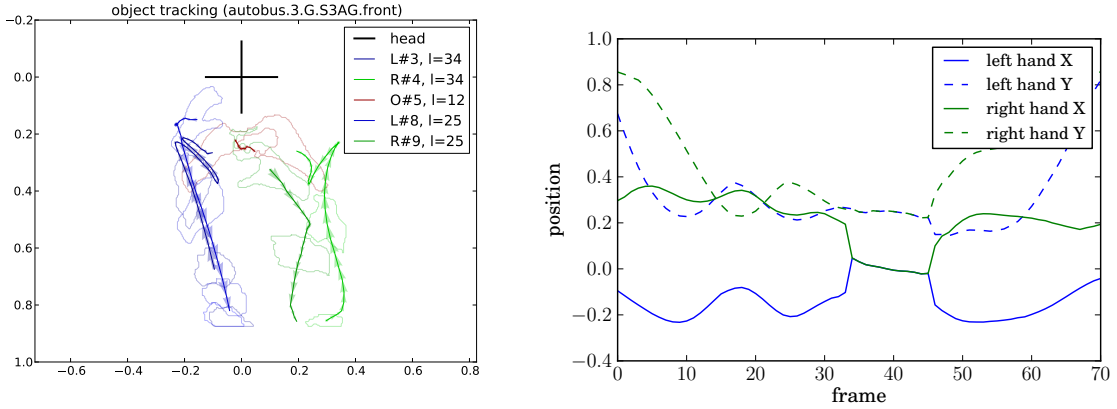


Figure 6.10: Examples of hand tracking, two-handed sign AUTOBUS with hand occlusion.

described in the following.

Hand Classification

Having two time aligned sequences of skin color regions as a result of the previous tracking algorithm, with a prior knowledge that one corresponds to the left hand and another to the right hand, the task is to associate each region with either left or right hand. Having two sequences and two hands, there are only two possibilities, either the first sequence is associated to the left hand and the second to the right hand, or vice versa. To solve this task a binary classifier and majority voting rule [KHDM98] are employed.

Several feature types were used for the description of the skin color region (see table 6.1). The feature vector used for the classification either consists of features calculated from one skin color region or from two regions where the individual feature vectors were concatenated into one single vector. The performance of both approaches was measured and the results are summarized in table 6.1. The column "single" corresponds to the first case where only features of the examined region are used for the classification and no information about the second region is used. The column "pair" is the second case, where features corresponding to examined region and the second region are concatenated into a single feature vector. Thus, the second hand adds more information about the context in which the first hand occurred (fig. 6.11).

The expectation to have higher accuracy for the classification based on observation of both hands was confirmed by the experiment. The evaluation dataset was automatically created by the tracking method applied to the UWB-06-SLR-A and UWB-07-SLR-P corpora with manual verification. The resulting hand corpora were described in section 6.1.

A classifier based on Support Vector Machines (SVM) and implemented by [Mil] is used for the supervised classification. This particular implementation uses a *Stepwise Discriminant Analysis* (SDA) dimension reduction method and nonlinear SVM [Bur98] with *Radial Basis Function* (RBF) as a kernel function. The parameters of RBF are obtained by a *grid search* algorithm that measures the classification accuracy for each considered combination of parameters and picks the combination that performs best.

The training set contained 2000 or 10000 randomly selected hand pair samples, denoted as N. The resulting classification accuracy is based on random permutation cross-validation with 3 repetitions.

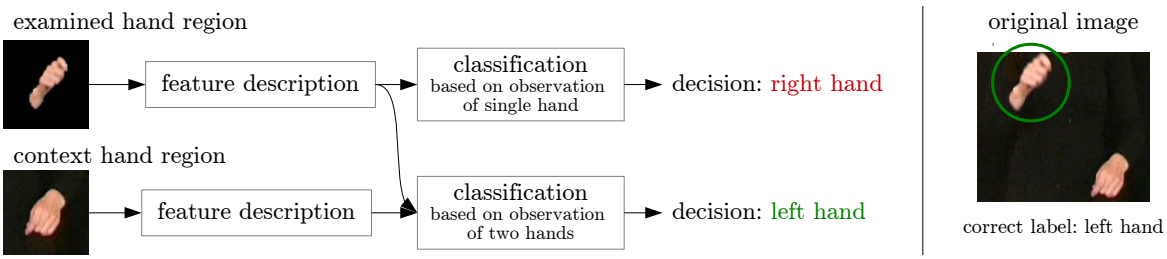


Figure 6.11: Example of left-or-right hand classification for two input images. First classificator uses only one image for the decision, the second uses both images.

	UWB-06-SLR-A		UWB-07-SLR-P	
	single	pair	single	pair
$N = 2000$				
normalized central moments η_i	65.70	52.85	51.05	69.71
Hu moments I_i	50.15	51.70	50.63	50.00
LBP (radius 2, 64x64px, uniform)	**77.90**	**82.60**	**80.27**	**83.55**
HOG (cell 8x8, 112x112px)	74.47	73.19	73.35	75.75
hRDF (72 angles)	74.84	78.21	70.60	73.79
x,y position	**88.92**	94.30	**95.96**	97.32
LBP + position	74.57	**99.31**	94.64	**98.45**
$N = 10000$				
LBP	**89.47**	**93.81**	**86.20**	**90.29**
LBP + position	**99.08**	**99.83**	**98.51**	**99.11**

Table 6.1: Hand classification accuracy [%]. "Single" column uses only features calculated from one hand region, "pair" uses features from both hand regions. N denotes number of samples in the training set.

The results in table 6.1 show that the best result was obtained for feature vector consisting of concatenation of LBP features and position features. The uniform LBP features were using radius 2, calculated from normalized hand image resized to 64x64 pixels. The position feature vector (x, y) is collected from normalized coordinate system that was described in section 6.1.1. The best classification accuracy was 99.83% for UWB-06-SLR-A hand corpus and 99.11% for UWB-07-SLR-P hand corpus.

For some applications it can be useful to classify the hand based only on features calculated from the hand region and without availability of the position. In such a case the best results were 93.81% for UWB-06-SLR-A and 90.29% for UWB-07-SLR-P hand corpus. This shows that the position is important information used for hand classification.

The classifier described above can be applied for hand regions that were segmented by the head and hand separation algorithm. It is expected that the image contains one or two hand regions only. The robustness can be increased by *majority voting* that classifies the hand tracked sequences as a whole.

The tracking algorithm guarantees that the tracked blob cannot be confused with another, thus the whole tracked sequence certainly contains the same hand. At first, the hand classifier is applied to each frame of the sequence. In some, the classification fails, as is depicted on an example in fig. 6.12. The frames where the classification results of both hand tracks were identical are ignored from further processing, because the classifier was confused in one of the tracks, thus this frame probably contain some peculiar hand configuration that was not seen during the training of the classifier. Then a simple majority voting decision rule is applied to both hand tracks. Thus, the association of tracked hand blobs to either left or right hand was found.

	majority voting result
hand blob 1: L L L L R L L L L R L L L L L L L L R R L L	L
hand blob 2: R R R R R L R R R L R R R L R R R R L L R L R	R

Figure 6.12: Example of majority voting used for hand classification of two time aligned hand tracks. "L" denotes classification of given blob as left hand, "R" as right hand. Identical classification results in the pair of hands are ignored from majority voting final decision.

6.2.3 Manual Component Features

This section describes several methods for feature extraction of manual sign component that were implemented and their performance was measured in the recognition process. A common goal is to describe a segmented hand region by a feature vector that can be used in later processing stage. The trajectories of the hand movements, which are a part of the sign manual component, are already known from the previous hand tracking step.

As the number of hands present in the signing space differs during the time, i.e. can be one or two, the question arises how to properly built feature vectors in both cases. When two hands are visible, the two resulting feature vectors are concatenated into a single feature vector, preserving the order of hands. In the second case, when only one hand is visible, there are several possible approaches. The basic approach is to use "empty" feature vector for hidden hand, e.g. to use vector filled with zeros, but this can cause some discontinuities of feature values in time when the hand enters or leaves the signing space. The approach employed in this work considers duplication of feature vector of the visible hand for the hidden hand, thus the feature vector of the hidden hand is the same as of the visible hand. This suppresses some discontinuities in the series of feature vectors.

Local Binary Patterns - LBP

Given an image with hand, the Local Binary Patterns (LBP) method is used to extract a feature vector from the source image that was normalized to size of 64 pixels. As was described in section 5.1.4, the LBP method has some parameters that must be manually adjusted:

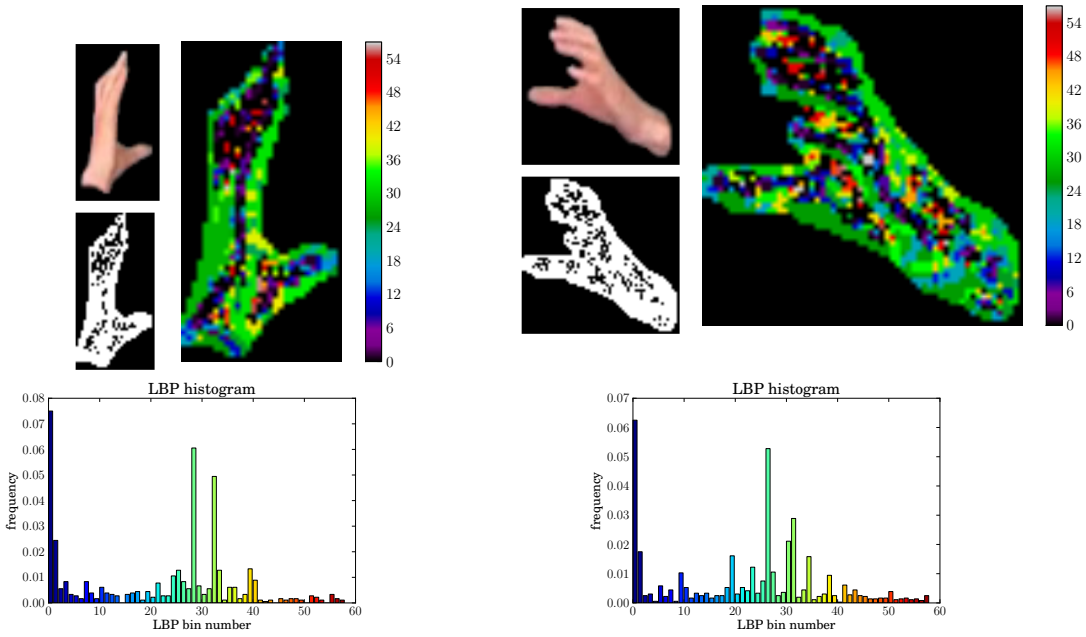
radius Values from 1 to 4 were examined. Generally, the best performance was obtained with values 2 or 3.

uniformity In all the experiments the uniform LBP variant was used. It is more robust and generates lower dimensional feature vector than the non-uniform variant.

neighborhood Fixed 8-neighborhood was used in all the experiments.

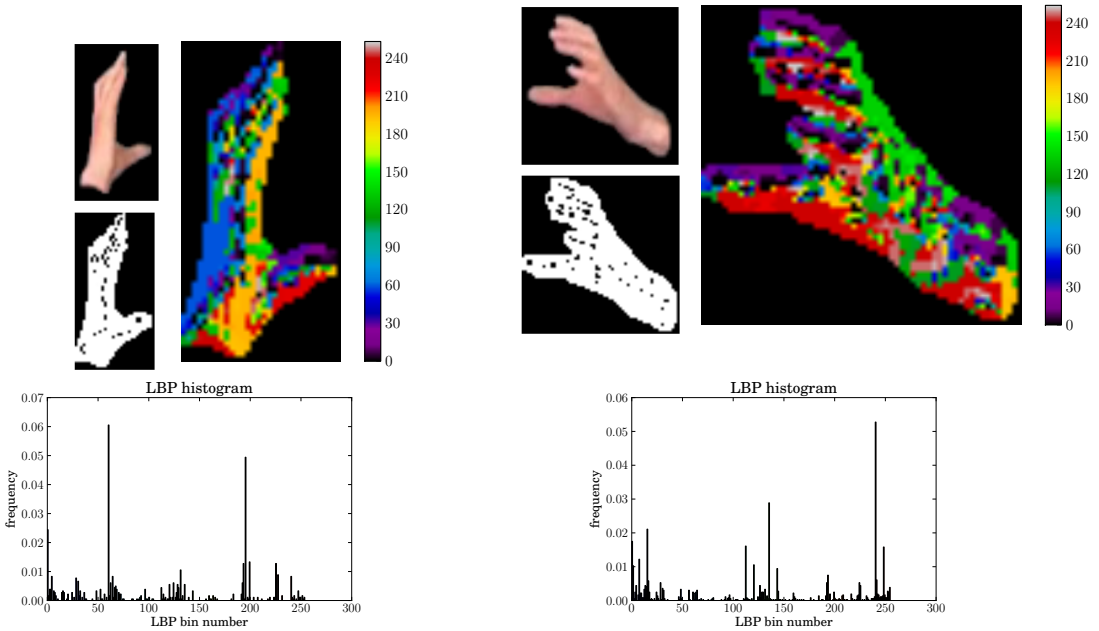
The figure 6.13 shows examples of LBP calculation for two different images. For each pixel in the image a pattern number (0 to 59) is calculated and is depicted by a different color. Additionally, a binary mask shows which pixels are used and which are ignored. The resulting histogram of the pattern numbers is shown at the bottom.

The same images are used in another example shown in fig. 6.14 with the same parameters except the non-uniform variant is used. The comparison of the binary masks shows that the non-uniform variant is ignoring a lower number of pixels, which is an expected behavior.



(a) Sample hand image 1 **(b)** Sample hand image 2

Figure 6.13: Examples of Local Binary Patterns calculation, uniform variant with radius 1 and 8-neighborhood. (left) sample image 1. (right) sample image 2.



(a) Sample hand image 1

(b) Sample hand image 2

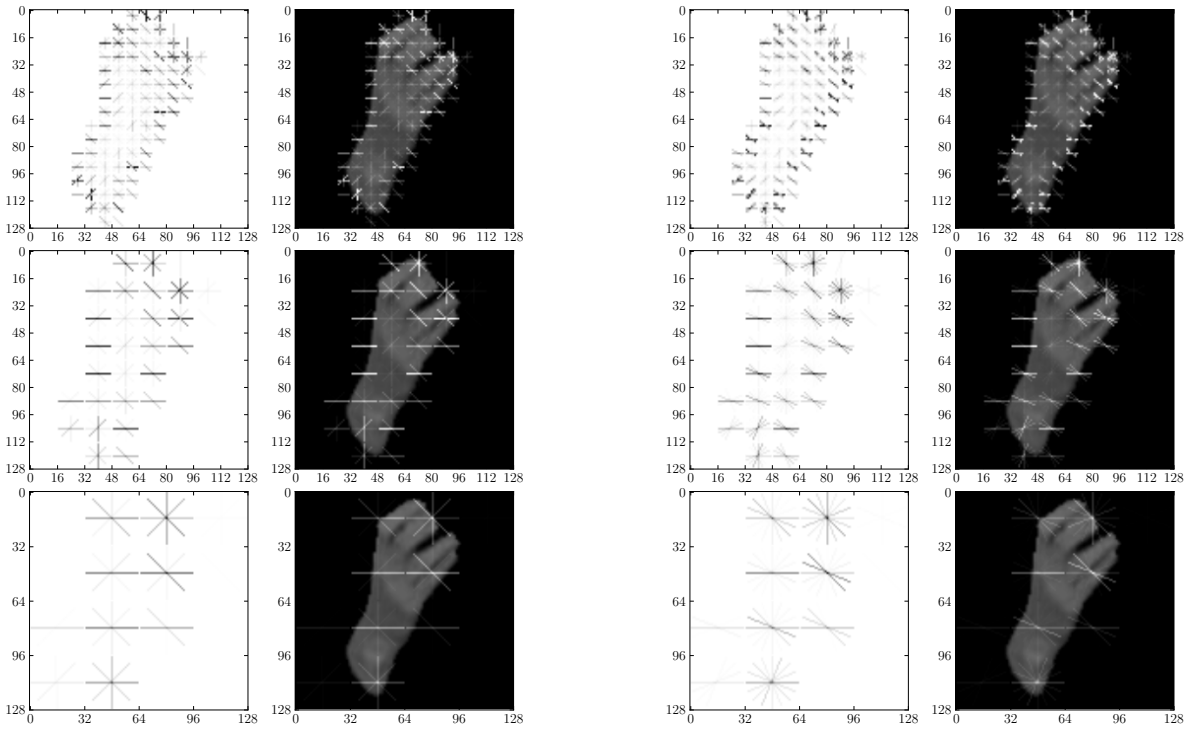
Figure 6.14: Examples of Local Binary Patterns calculation, non-uniform variant with radius 1 and 8-neighborhood. (left) sample image 1. (right) sample image 2.

Histogram of Oriented Gradients - HOG

This method, described in section 5.1.4, was employed in analogous way as previously described LBP for hand feature extraction. The parameters that were examined are: 4 or 8 for number of orientations, and 8 or 16 for the cell size. The block size was fixed to size 1x1 which results in the lowest possible dimensional feature vector, but without robustness in different light conditions, which is not needed for the used data. The source image was normalized to the size of 64 pixels.

The examples in figure 6.15 show different HOG results for different number of orientation and cell size values on 128px image.

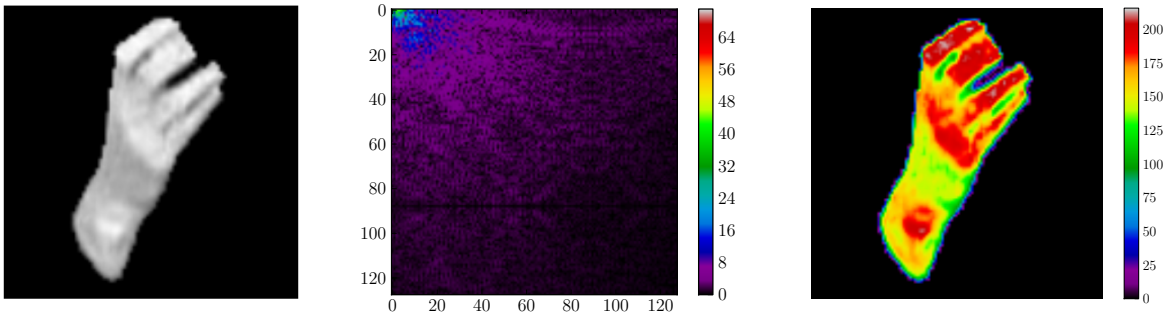
Additionally to hand feature extraction, the HOG feature descriptor was applied to the full normalized source image with whole signer, containing hand and head blobs. This approach can be used directly without the need to process tracking, but it did not reach top performance as can be seen later.



(a) 4 orientation bins

(b) 8 orientation bins

Figure 6.15: Histogram of Oriented Gradients: gradient directions for cell size 8x8 pixels (top), 16x16 (middle) and 32x32 (bottom). The left images show the resulting gradient directions, the right images show the directions in overlay with the source image.



(a) Example image **(b)** DCT coefficients **(c)** Inverse DCT

Figure 6.16: Example of DCT and inverse DCT calculation.

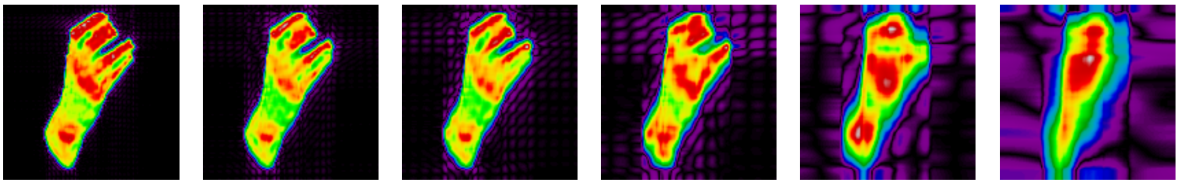


Figure 6.17: Inverse DCT using a $N \times N$ subset of DCT coefficients. $N = (30, 20, 15, 10, 5, 3)$. The grayscale images are visualized using a color map for a better insight.

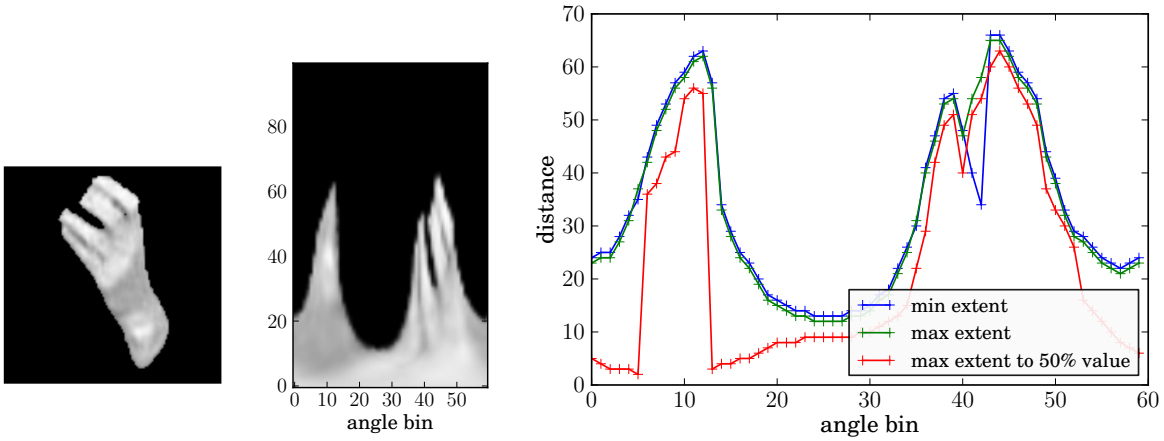
Discrete cosine transform - DCT

Discrete cosine transform (DCT) can be applied to the normalized input image and the resulting subset of DCT coefficients forms a feature vector. Such an approach is often used for mouth feature description [CÓ6]. The DCT coefficients with the highest values correspond to the low spatial frequencies which are visualized in the fig. 6.16b in the top left corner of the matrix. Thus, when only a subset of the DCT parameters is used, only the top left values are selected, either in a zig-zag manner or as a top left submatrix, which is used in this work.

The figure 6.17 shows several examples of inverse DCT calculated from a subset of DCT coefficients, when a top left submatrix with size $N \times N$ is used. The examples show that using a submatrix with size about 10×10 holds enough information about the hand shape for the recognition tasks.

Hand Shape Radial Distance Function - hRDF

An extension to original RDF method (section 5.1.4), denoted as hRDF, is proposed in this work with a goal to enhance the feature description of hand-shape like concave regions. In addition to RDF, hRDF measures not only maximal extent from the centroid of the region to the silhouette contour in several uniformly distributed directions, but adds additional measurements. The first additional measurement is minimal extent from the centroid to the contour. This value is the same as maximal extent in directions, where no concave region penetrates the object. Another case, which is common in the finger region, as can be seen in figure 6.18c, the *maximal extent* value differs from *minimal extent*.



(a) Example image

(b) Image mapped to polar space.

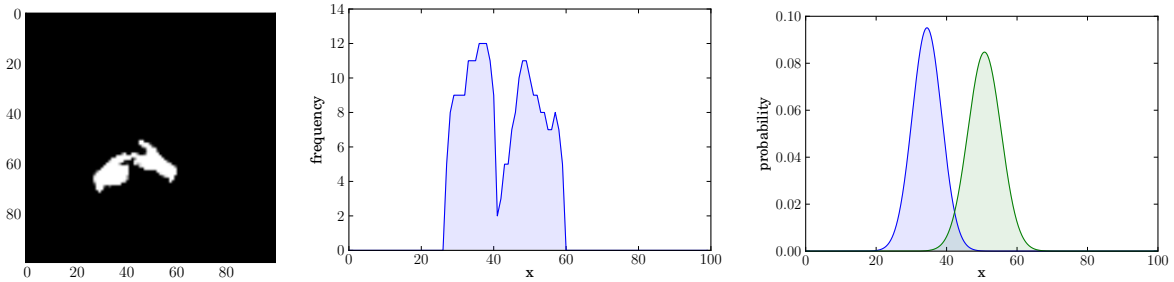
(c) Measurements from the polar space image, for each angle: minimal extent, maximal extent and maximal extent to 50% value pixel.

Figure 6.18: Hand Shape Radial Distance Function - hRDF

Until now, only the shape contour was described, without considering the pixel values inside the region. The second additional measurement is the maximal extent from the centroid to the pixel with value higher than 50% of the maximal pixel value in the region. This allows to roughly describe the distribution of "bright" pixels in the region, as is shown in the fig. 6.18c as *maximal extent to 50%*. The resulting feature vector is a concatenation of the three described measurements.

Radon Transform and GMM

Another hand shape descriptors proposed in this work are based on Radon transform where a two-dimensional binary image is projected into one dimension in different directions. A sum of pixels is calculated over straight lines in each direction. In the example image with hand mask image in figure 6.19a, the sums of pixel values in direction 0° are calculated and the



(a) Example hand mask

(b) Histogram of pixel values, direction 0°

(c) Distribution modeled by a GMM with 2 mixture components

Figure 6.19: Example of Radon transformation, its histogram and modeled distribution.

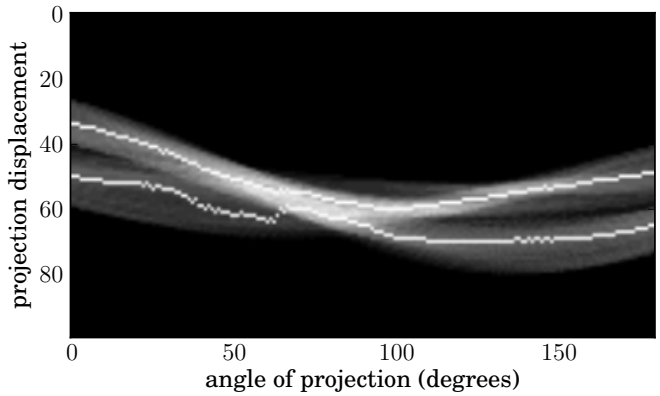


Figure 6.20: Radon transformation for different angles of projection. The white points denote the mean values of GMM that was used for the distribution modeling in the given direction.

results are shown in figure 6.19b. In this direction, the result can be interpreted as a histogram of pixel distribution in x axis.

The first considered feature descriptor based on the Radon transformation uses 4 different directions and the number of histogram bins is reduced to 20. This allows describing the pixel distribution in the source image.

The second considered feature descriptor is based on the knowledge that maximal number of regions in the image is two. The distribution of values in the histogram is modeled by Gaussian Mixture Model (GMM) with 2 mixture components. For each angle used, the GMM describes the distribution by a 6 dimensional vector, consisting of two mean values and four values from a covariance matrix. These GMM parameters are estimated by the expectation-maximization (EM) iterative algorithm and are used as the feature descriptor, denoted as *radon+gmm* features. An example is shown in figure 6.19c, where the distribution of the values in the histogram is modeled by a GMM with two mixture components.

The figure 6.20 shows the results calculated for all directions between 0°and 180°, from the same source image (fig. 6.19a).

Although this feature descriptor discards a lot of information from the image, it has a nice property that it can separate two occluded hand regions, as was seen in fig. 6.19a, by estimating the center positions of each hand as the mean values resulting from the GMM model.

Other Basic Descriptors

To fully compare the recognition results of different hand feature descriptors, some other commonly used descriptors were considered.

A combination of simple scalar region descriptors was used, consisting of *angle* (angle of the longer side of a minimum bounding rectangle), *solidity* (ratio of region area and convex hull area), *extent* (ratio of area and bounding rectangle area), *ratio* (ratio of width and height of the region).

Two other feature descriptors based on moments were used as well. The first uses a set of normalized central moments, the second uses *Hu moments* consisting of 7 rotation, translation

and scale invariant moment characteristics [SHB07].

6.2.4 Non-manual Component Features

The non-manual component of a sign that is contained in the face expression is modeled by the Active Appearance Model (AAM) described in section 5.1.5 and an example of model fitting was shown in figure 5.11. The AAM was trained from UWB-SLR-07-P dataset only, because the training stage of AAM require huge amount of manual work that is out of the scope of the thesis. It was verified that such a model fits other faces from UWB-SLR-06-A corpus in sense that the spatial placement of the model vertices is correct, but the appearance modeling has some inaccuracies in case the appearance of the modeled target face is too different from the four faces in the training set. For example, the appearance of a male face with a beard is not modeled well, because the model was trained on four female faces.

As was theoretically described, the property of the AAM is that the fitting can fail in case of occlusion. This was solved by the previously described head and hand separation algorithm (section 6.2.2) that approximates the face region by a template in case of occlusion. Thus, the face region is never occluded and AAM fitting is seamless and robust.

The first feature descriptor that was based on AAM is a direct usage of fitted AAM parameters. The disadvantage is that some parameters describe the face appearance, which is not desired for signer independent face description.

The second feature descriptor, denoted as *aam-ext*, removes these signer dependent parameters. From the AAM parameters, several distance measurements from shape model (the triangular mesh) are performed and used in the resulting feature vector. The distances measured are: height and width of the outer and inner lips, and distance between eyelids for each eye. Together, the resulting feature vector includes 6 parameters only. The advantage is that such measurements are almost signer independent and the size of the feature vector is much lower than of the first AAM feature descriptor.

6.3 Sign Language Recognition of Isolated Signs

In this section, the proposed sign language recognition system is described. As was already shown in figure 5.1, the system consists of four main modules. *Video analysis* is used for feature extraction from sign language video, and is described in section 6.3.1. *Sign modeling* module, that calculates an estimation of the likelihood $p(W|\boldsymbol{O})$ is described in section 6.3.2.

Language modeling uses uniform language model for the experiments, thus it is not described more in detail. As was already discussed, the collection of the data required for language modeling of sign languages is a difficult task and is not aimed by this work.

The last module containing a *decoding algorithm* was described in section 5.4. The available implementation in *HTK toolkit* [YEG+06] was used as a reliable and proven implementation that is widely used in speech recognition tasks. The same toolkit was used in the sign modeling module, where both modules are tightly connected and optimized.

6.3.1 Video Analysis

The goal is to analyze both manual and non-manual components of the performed signs that are present in video sequences. The result of the analysis is a feature vector extracted from each video frame. The vector can be produced by a fusion of multiple feature vectors that were calculated by different methods.

The implementation used for non-manual component analysis through AAM was already described in section 6.2.4.

The manual component analysis was performed by several methods that were described in section 6.2. The methods with adjustable parameters were evaluated with multiple parameter settings with a goal to find the best parameters fesible for the sign recognition task.

All the input frames are normalized and preprocessed by the skin-color segmentation method. The performance of low-level methods that use whole image for feature extraction, such as eigensigns, was compared with high-level methods that employ head and hand tracking and features are extracted from hand regions. The results are presented in section 6.3.3.

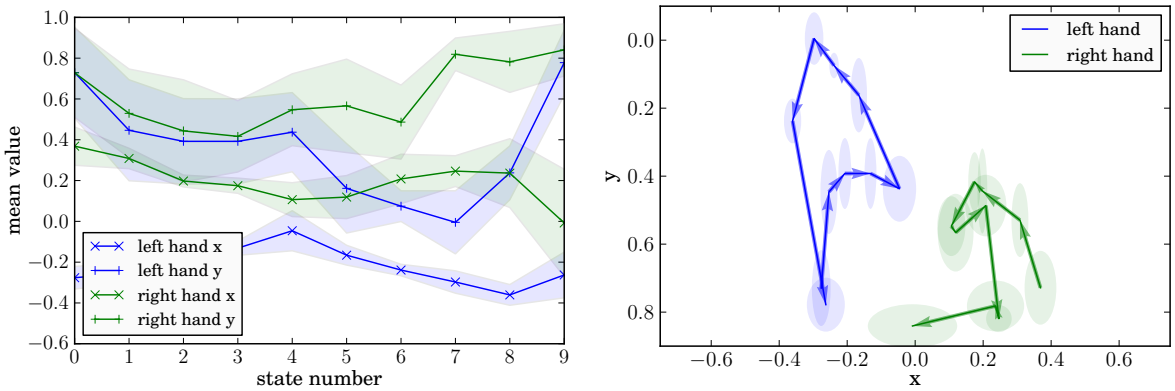
The dimensionality and correlation of feature vectors was reduced by PCA (section 5.1.6) where it was suitable.

6.3.2 Sign Modeling

The purpose and functionality of the sign modeling module was described in section 5.2. In this work, Gaussian Mixture Model (GMM) was used as the observation probability function. The optimal number of the mixture components was estimated from recognition results where several numbers were evaluated, independently for each feature descriptor.

Number of HMM states used for sign modeling was estimated in a similar way and was the same for all signs. The allowed transitions from i-th state were only to the same state or to the subsequent i+1 state, such a transition network was depicted in figure 5.13.

The figure 6.21 shows an insight to a particular HMM model, where the (x,y) coordinates of both hands are used as a feature vector. Such features allow meaningful visualization. Each



(a) Mean values and variances for each state

(b) Interpretation of the modeled coordinates in the signing space

Figure 6.21: Example of a HMM model associated to sign AUTOBUS, with 10 states, using (x,y) coordinates of both hands as feature vector.

of the 10 depicted HMM states models the distribution of each coordinate by a GMM, where the mean values and variances are depicted in the graph.

In order to improve the recognition performance, *dynamic features* that incorporate temporal derivatives of the feature vectors and are widely used in the field of automatic speech recognition, can be computed from any kind of features and used for the recognition. The dynamic features can measure changes of the original features and can improve recognition performance for some types of features. The dynamic features computed as first order time derivatives are denoted as Δ features, velocity parameters, or *delta coefficients*. The second derivatives are denoted as Δ^2 features, acceleration parameters, or *delta delta coefficients*. Generally, the Δ features can give a large gain while the Δ^2 features add a smaller one. In this work, Δ^2 symbol denotes both velocity and acceleration parameters, while Δ denotes only velocity parameters. For example, the figure 6.21b represents a sign model, having (x,y) coordinates as features. The model represents only positions in the signing space in a time sequence. With Δ features, the model would represent the movement velocities and directions too, which is intuitively an important feature that enhances the model and improves the recognition performance.

6.3.3 Experiments

Both corpora UWB-06-SLR-A and UWB-06-SLR-P were used for the experiments. The first contains higher number (15) of signers, but only 25 different signs. The second was contains only four signers, but 378 different signs. 336 of signs were selected for the recognition experiments, the remaining 42 signs were unsuitable for this experiment setup, for example some of those signs were performed incorrectly by some speakers.

For each experiment, the used dataset was split into exclusive training and testing subsets. The training set was used for training of HMM models and the testing set was used after for the evaluation of the recognition performed on these HMM models.

For the signer dependent recognition, the items in the subsets are randomly selected from the whole dataset. To estimate a suitable ratio between the number of items in training and testing sets, a minor experiment that measured the recognition accuracy in dependence on the ratio was performed on UWB-06-SLR-A corpora with *radon+gmm* features. The graph 6.22 shows the recognition accuracies with 95% confidence intervals that were estimated by the bootstrap method (described in section 5.4.2). It is evident that with more training data used, the recognition accuracy is higher, but for the price of less confident accuracy measurement due to low testing data quantity. As is shown in the figure, the confidence interval is wider for lower number of testing data. The proportion that was selected for all further experiments is 25% for the testing data and 75% for the training data. This ratio showed nearly the same recognition accuracy, but had higher confidence than the ratios with lower number of testing data.

For the signer independent recognition, the data of one signer are used for testing and the rest of signers is used for training.

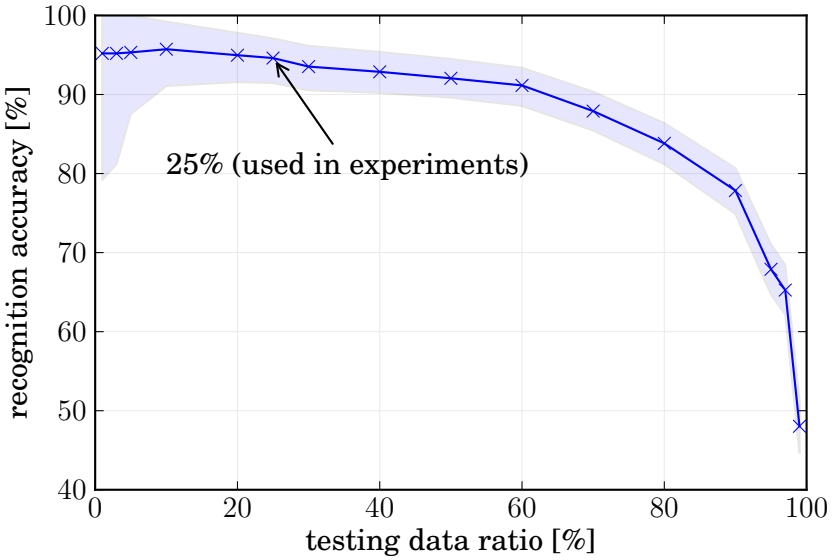


Figure 6.22: Recognition accuracy and its 95% confidence interval depending on the training and testing data ratio. Experiment was performed on UWB-06-SLR-A corpora with *radon+gmm* features.

recognition accuracy [%]				feature descriptor
signer dependent		signer independent		
SLR-A	SLR-P	SLR-A	SLR-P	with hand tracking
95.73	84.47	89.69	29.46	HOG hand features
93.03	83.51	92.12	26.87	(x,y) hand coordinates + Δ^2
91.69	-	82.40	-	(x,y) hand c., no head/hands separation + Δ
91.01	76.44	86.71	20.16	hRDF hand features + Δ
86.07	57.16	90.73	18.92	high level linguistic features
74.61	55.15	75.19	13.93	Hu moments
				without hand tracking
98.65	**93.19**	94.23	27.85	LBP + Δ^2
86.18	72.69	76.08	14.36	DCT
85.53	71.12	75.16	13.95	radon transformation
-	47.78	-	-	eigensigns
74.05	39.59	70.83	7.10	pixel values as features
46.87	43.24	12.27	1.43	AAM
29.37	24.47	8.32	1.34	AAM-ext
				fusion of multiple features
95.51	91.54	**95.06**	32.47	(x,y) hand coordinates + LBP hand features
95.06	84.64	95.05	**34.47**	high level linguistic features + LBP hand features
95.73	84.21	90.80	29.46	(x,y) hand coordinates + HOG hand features
95.28	80.63	92.07	29.82	high level linguistic features + HOG hand features

Table 6.2: Accuracies for recognition of isolated signs. UWB-06-SLR-A corpus is denoted as SLR-A, UWB-07-SLR-P as SLR-P.

All the results presented in the table 6.2 are based on random permutation cross validation with 3 repetitions, i.e. every experiment was repeated 3 times based on different random selection of testing and training data, the final result is an average value of particular results. The table summarizes experiments performed on both corpora and both signer dependent and independent results. The feature extraction methods listed in the table are the best performing from each group and their exact settings are discussed in the following. The first part of the table presents recognition accuracies for feature extraction methods that employ hand and head separation and hand tracking. The second part of the table presents methods where no tracking was employed. The last, third part, addresses results for fused features from multiple feature descriptors.

All methods were benchmarked in three different settings: without delta coefficients, with delta (velocity) coefficients Δ and with delta delta (velocity and acceleration) Δ^2 coefficients. All possible combinations of HMM parameters were benchmarked, namely number of HMM

states (varying from 3 to about 13) and number of GMM mixture components (from 1 to about 13). Only the best performing configurations are presented in the table 6.2. Extended table that includes these HMM parameters is in appendix A, together with detailed parameters of particular feature extraction methods.

As was expected, the recognition accuracy is highly dependent on used feature descriptor type. The interpretation of the results is that the best performing feature descriptors are based on LBP, either applied to the whole image containing hand regions (denoted as "LBP") or applied to each hand separately (denoted as "LBP hand features"). The accuracy for the signer independent recognition was increased when a fusion of multiple feature descriptors was used, in particular "LBP hand features" with either "hand coordinates" or "high level linguistic features". In general, the experiments that make use of higher level features performed better, the accuracy of low level methods ("eigensigns" and "pixel values as features") was much lower.

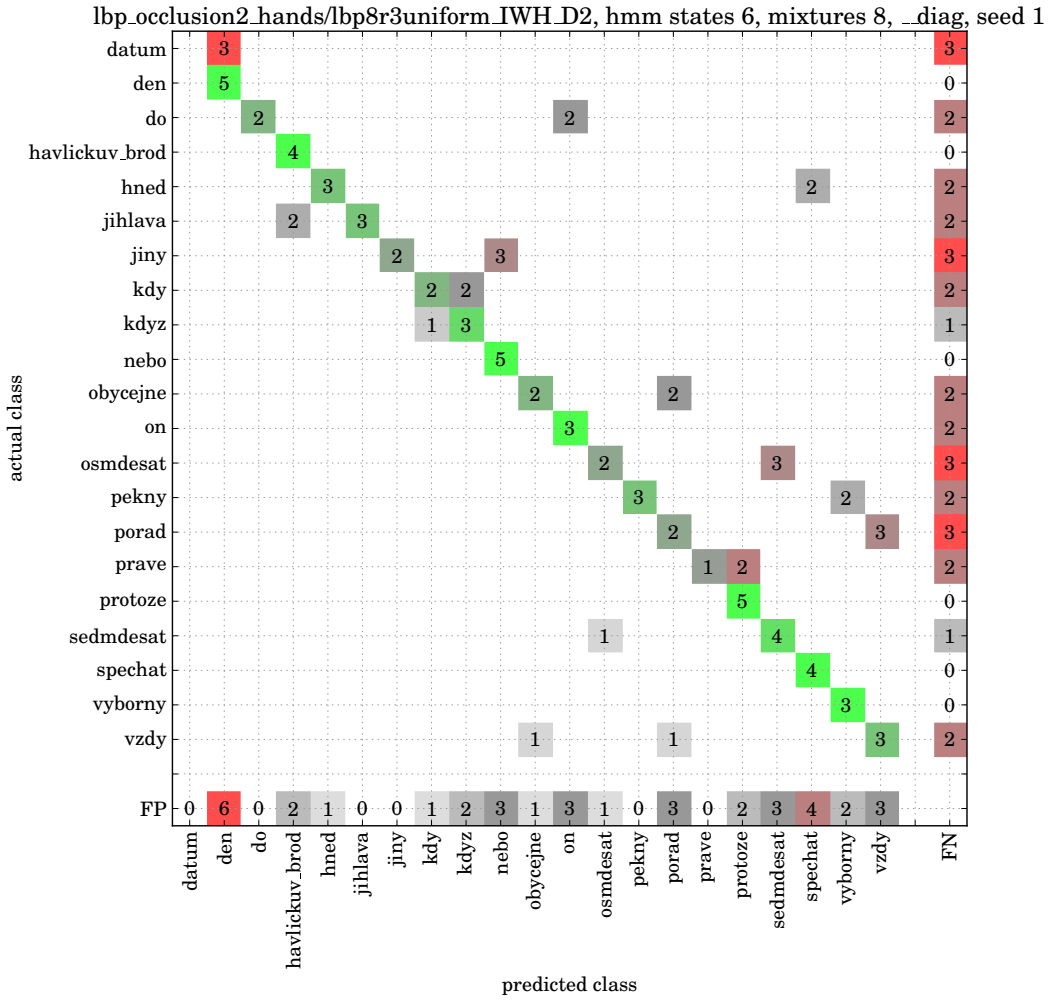


Figure 6.23: Selected rows and columns from confusion matrix for most confused signs. Experiment on UWB-07-SLR-P corpus, LBP + Δ^2 features, HMM 6 states, GMM 8 mixture components. FP denotes false positives for given sign, FN false negatives.

The experiments with the standalone "AAM" and "AAM-ext" features use only non-manual component of the sign for the recognition (the manual component is ignored), so the recognition accuracy is expected to be low. Nevertheless, the recognition rate shows that the non-manual features contain some information that can be utilized for discrimination of the signs. Some fused combinations of AAM features with other manual component features were used in other unlisted experiments (AAM-ext + LBP, AAM-ext + high level linguistic features), but the accuracy was slightly lower in comparison to the same features without AAM-ext. A question arises why the use of non-manual face features reduced the recognition accuracy. One possible explanation is that the face features were collected from all frames of the video and that some of them contain more face movements that are not related to the sign, thus these movements act as a noise. Additional issue is the resolution of the face region that can be insufficient.

Since the UWB-07-SLR-P corpus contains more signs than UWB-06-SLR-A, the recognition accuracy is lower for the same experiments. By deeper inspection of the confusion matrix in figure 6.23, where only rows and columns of the full confusion matrix corresponding to the

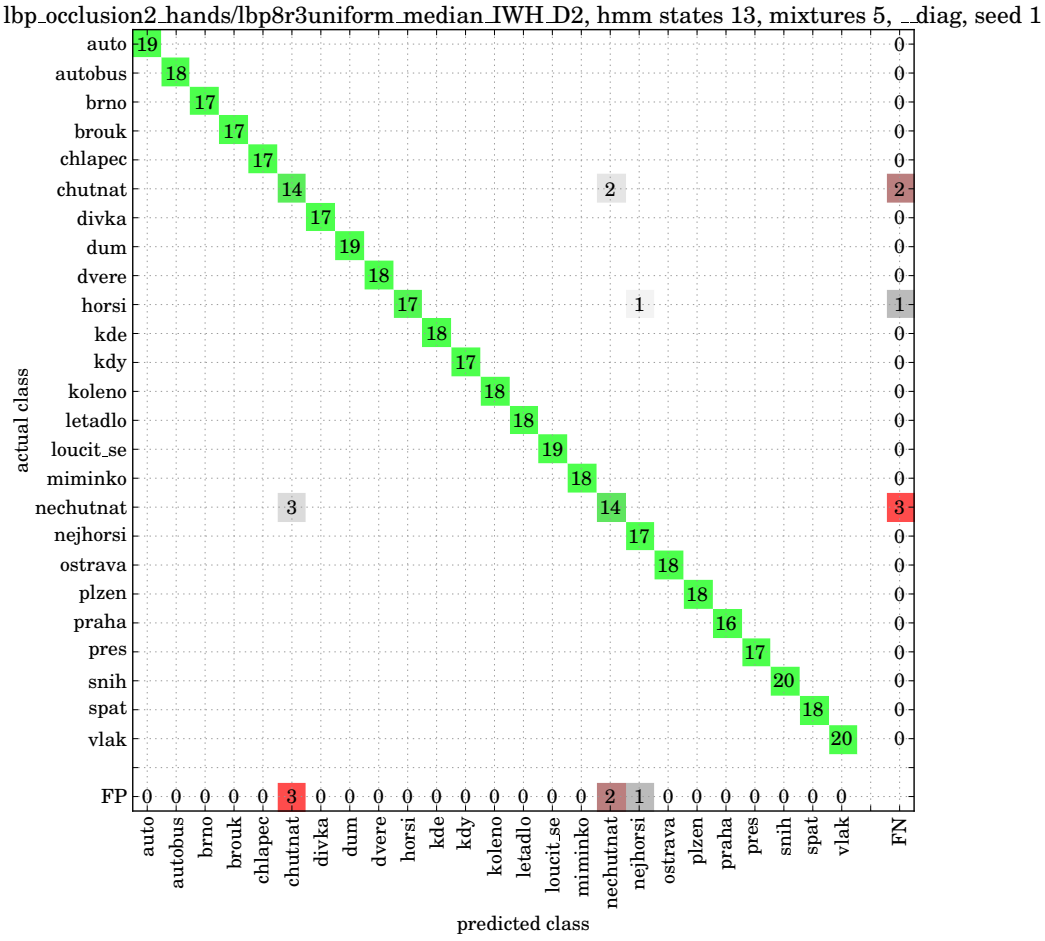


Figure 6.24: Full confusion matrix. Experiment on UWB-06-SLR-A corpus, LBP + Δ^2 features, HMM 13 states, GMM 5 mixture components. FP denotes false positives for given sign, FN false negatives.

most confused signs are shown, some other conclusions can be stated. Some signs are confused (DATUM (date) vs DEN (day), DO (to) vs ON (he)) because the manual component is the same or very similar and the sign differs only in non-manual component. The fusion with AAM features can resolve some confusion in such a case, but causes some other and the total accuracy is lower with AAM. Other signs like SEDMDESAT (seventy) and OSMDESAT (eighty) are confused because the hand configuration differs only in one finger and the feature descriptors are not discriminative enough to resolve this small difference.

The signer independent accuracies are lower than for signer dependent, as expected. The results for signer independent experiments on UWB-07-SLR-P corpus dropped rapidly in comparison to UWB-06-SLR-A. There are several possible reasons. The first is that for better performing UWB-06-SLR-A, the training of the recognition system was based on data from 14 signers and the recognition on one signer. On UWB-06-SLR-A, only data from three signers were used for the training. The next reason is that some signs from the same class were performed differently by each signer, although a subset of the corpus was selected to avoid such cases, for some signs this was overlooked and caused that the sign model was trained and tested on differently performed signs.

A full confusion matrix for the best performing result on UWB-06-SLR-A corpus with LBP + Δ^2 features is depicted in figure 6.24. All confusions are for the sign pairs which differ only in non-manual component of the signs.

Last experiment, presented in figure 6.25, demonstrated the use of PCA feature reduction method, again on the best performing result on UWB-06-SLR-A corpus with LBP + Δ^2 features, where the original feature vector has dimension 177. The experiment shows that the recognition accuracy remains high even when the dimension is highly reduced. For example, when using only 20 dimensions instead of original 177, the accuracy drops from 98.65% to 97.3%. With such a reduced feature vector size, the training and recognition process gains some speed.

6.3.4 Towards the Creation of Data-driven Sub-units

As was introduced in the section 5.2, the concept of *sub-units* uses HMM models representing units smaller than whole sign. The sign is represented as a concatenation of several sub-units, which can be shared among multiple signs. The sub-units used for sign language modeling are usually data-driven, i.e. are constructed by an analysis of the training data since no linguistically proposed sub-units are available.

We proposed an unsupervised iterative method that serves as a first step for data-driven construction of sub-units. The method employs Gaussian mixture models (GMM) (section5.2.1) which has been well studied and examined in similar tasks in the field of automatic speech recognition. The idea is to create a set of Gaussians, a *pool*, which models the distributions of all the data present in the training set. Then, HMMs are built for each sign as a linear combination of Gaussians from the pool. Thus, the HMM for each sign is fully defined only by a weight matrix W, that contains weights representing the linear combination of the Gaussians for each state. The number of states is estimated too and can be different for each HMM. Similar approach was studied in [VC99] for the field of automatic speech recognition.

The result of the proposed method is a set of HMM models, each modeling a single sign. The

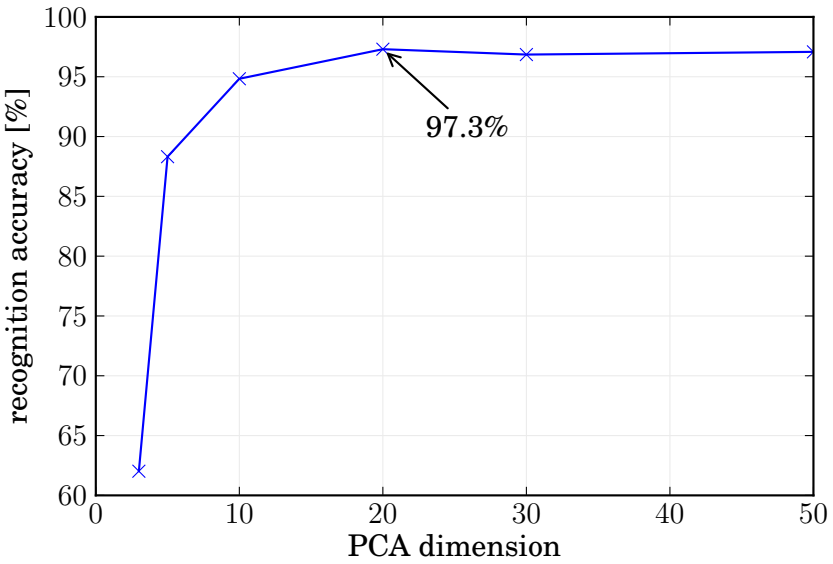


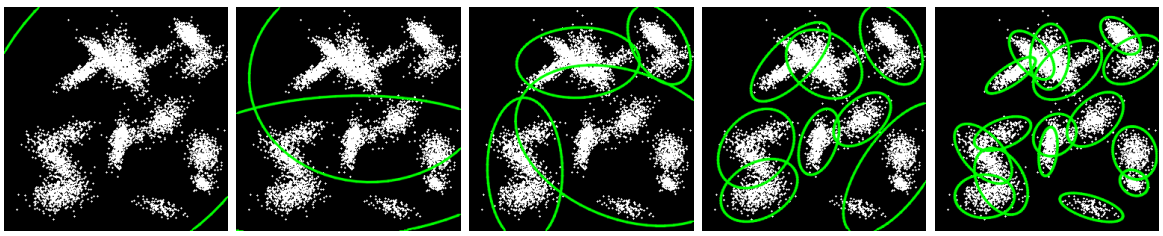
Figure 6.25: Example of PCA feature reduction method usage. Experiment on UWB-07-SLR-P corpus, LBP + Δ^2 features, HMM 6 states, GMM 8 mixture components.

difference from the HMMs that were used in this thesis for isolated sign language recognition is that the observation probabilities are now modeled as a linear combination of Gaussians, that are shared among all models.

The second step that creates proper sub-units was not realized in the scope of the thesis and is left for future work. A clustering method is needed to identify similar HMM states among all HMM models, i.e. states that have similar weights of Gaussians. After that, sequences of the same states among all HMM models can be grouped and create sub-units as sequences of HMM states, so that such a sequence can be used as a standalone HMM model that represents a single sub-unit.

The proposed method consists of 5 steps:

1. **Gaussian mixture pool generation**
   The training data in form of time series of feature vectors are sequentially modeled



(a) 1 component **(b)** 2 components **(c)** 4 components **(d)** 8 components **(e)** 16 components

Figure 6.26: Example of a Gaussian mixture pool generation. The images present GMMs of sample 2D training data using 1, 2, 4, 8 and 16 components (Gaussians). The resulting pool is a set of all 31 individual components.

by 1, 2, 4, 8, ..., 64 and 128 mixture components (Gaussians). The set of all 255 individual components is denoted as a *pool*. The distributions of the GMMs are estimated by the expectation-maximization (EM) algorithm. A two dimensional example of such a pool generated on 2D sample data is depicted in fig. 6.26, where the first image shows estimated GMM with one component that models global distribution of data in the feature space and the last GMM with 128 components that models distributions in smaller clusters. A question arises, why the pool is built from components of multiple GMMs and not only from the one with the most of the components. The answer is that at the beginning it is not known whether the observation distributions will be "general" or "specific". Thus, the method itself will choose which kinds of offered Gaussians are needed, depending on the training data.

2. **Initialization of HMM models**
   One HMM model is associated to each sign. The number of states is selected to be long enough. In the previous experiments it was shown that the sufficient number of states is below 15. Here, the length of 20 was selected, including special states 0 and 19 that are shared among all HMMs and are denoted as *S/E*, that models "start/end" events that appear before and after the sign. The i-th HMM model of the i-th sign is represented by a weight matrix $W_i = [w_{m,n}]_{m=1,...,255;n=1,...,20}$, where row m denotes m-th component from the pool and column n denotes n-th HMM state, thus the matrix element $w_{m,n}$ is a weight of m-th component associated to the n-th state. The sum of weights adds up to one in each state. The weights are initialized uniformly with the same values.

3. **Estimation of number of HMM states**
   In each iteration, the weight matrix W is updated, following the maximum likelihood criterion. The forward-backward algorithm is employed and counts, which of the Gaussians from the pool were used in given HMM model. In this step, the left-right HMM topology is used, but additionally transitions from each state to the last 19-th state is allowed. The number of iterations was fixed to 5. After the iterations are finished, the transition probabilities from i-th state to the last 19th state are compared and the state associated with the largest transition probability denotes new last state, that will be used for given sign and the rest of the states is truncated, except of the last *S/E* state. Thus from now, each HMM model has different number of states, lower or equal to the initial number of 20 states. The purpose of this step is to estimate optimal number of states required for each sign.

4. **Weight matrix update**
   In this step, five iterations as in the previous step are used to update the weight matrix W, but now the transitions are allowed only from i-th state to i-th or $(i+1)$-th state, no direct transitions to the last state are allowed. The purpose of this step is to update weight matrix W having the number of HMM states fixed.

5. **Weight matrix and GMM pool update iteration**
   This last step uses 5 iterations to update the weight matrix W, in the same way as in the previous steps, but additionally the Gaussians in the pool are updated too, so that the final Gaussians fit better the training data.

The results evaluated on UWB-06-SLR-A corpus are shown in the table 6.3. The "baseline" column evaluates the proposed algorithm without the last "W and GMM pool update iteration" step, thus the Gaussians in the pool are fixed. The "extended" column is a full version of the algorithm including the GMM pool updates. For comparison the third column "original" adds results from the same experiment performed with HMM models that were presented in section 6.3.3.

recognition accuracy			feature descriptor
baseline	extended	original	
72%	73%	97.3%	LBP + Δ^2, PCA 20 dimensions
74%	**77%**	96.9%	LBP + Δ^2, PCA 30 dimensions
73%	74%	91.7%	(x,y) hand coord., no head/hands separation + Δ

Table 6.3: Accuracies for recognition of isolated signs using sub-units.

To conclude, the proposed algorithm showed promising results and the possibility that the sub-units can be identified from the data. The best achieved recognition accuracy was 77%, which is lower than 96.9% achieved by the previous method.

6.4 Sign Language Recognition of Continuous Speech

The same system that was presented for sign language recognition of isolated signs, with HMM models that use one model per sign, is used here for experiments with continuous speech. The only difference is in language model that allowed only one sign sentences in the isolated case. For the continuous speech a uniform language model is used, that allows to build sentences of arbitrary lengths.

The HMM models are the same that were trained for isolated signs from UWB corpora. The recognition accuracy was evaluated on utterances that were randomly generated from isolated signs. For a given number of signs the utterance was generated by concatenation of randomly selected single signs, where their feature vectors, already computed in the previous step of isolated recognition, were concatenated. Thus, an unlimited number of artificial random utterances can be generated. The system was evaluated on 1000 sentences of lengths uniformly distributed between 1 and 20 for sign dependent experiments. Sign independent recognition was evaluated on 100 random sentences per signer.

This experimental setup allows performing experiments on continuous recognition although no continuous data are really available, but some aspects of the sign languages are ignored, such as coarticulation effects, where each sign is influenced by neighbouring signs in the sentence.

Two best performing feature descriptors from the isolated recognition experiments were used for continuous experiments. The results are shown in the following table 6.4.

accuracy [%] / correctness [%]				method
signer dependent		signer independent		
SLR-A	SLR-P	SLR-A	SLR-P	
97.19 / 97.57	86.00 / 90.91	86.89 / 88.27	19.79 / 24.78	LBP + Δ^2
91.45 / 95.01	74.50 / 78.95	86.30 / 90.40	13.32 / 34.40	high level linguistic features + LBP hand features

Table 6.4: Recognition results for continuous sign recognition. UWB-06-SLR-A corpus is denoted as SLR-A, UWB-07-SLR-P as SLR-P.

The accuracy evaluation uses accuracy and correctness, as described in section 5.4.1, where correctness is similar to accuracy but ignores word insertion errors.

For the signer dependent recognition on UWB-06-SLR-A corpus the accuracy computed on LBP + Δ^2 features was 97.19% which is similar to the result in isolated recognition with 98.65%. In this case, the proposed system is able to identify number of signs in the utterance, find their borders and correctly recognize the signs. With the UWB-07-SLR-P corpus that has more signs, the recognition accuracy dropped from 93.19% in isolated case to 86.00% in the continuous case.

For the signer independent recognition, the best accuracy achieved on UWB-06-SLR-A corpus was 86.89% (95.05% was for isolated case). The most difficult experiment, signer independent recognition on UWB-07-SLR-P corpus, where the accuracy in the signer dependent case was already low (34.47%) dropped to 19.79%. The reasons are the same as was discussed in the recognition of isolated signs. There are several possible ways how to increase the accuracy: to use other feature extraction methods and their fusion, use adaptation techniques to update the sign models for each signer, or use a language model better than the uniform language model. This case of signer independent recognition on corpora with more than tens of classes is similar to the field of automatic speech recognition where all the subtasks are still under research.

As was already explained, the utterances for continuous recognition experiments were artificially generated by plain concatenation of single signs, where each sign starts and ends in the neutral position. This makes the recognition task easier than in the real case where the sign are performed continuously and with coarticulation effects.

An experiment that tries to remove the influence of the neutral position effect is presented here. As was stated in [Ten10], "*the results show that the stroke alone performs as well as the entire sign*". This indicates that the stroke (main "central" part of the sign) contains enough information that can discriminate the signs. The proposed experiment measured the recognition accuracy of randomly generated utterances, where the signs were truncated, i.e. a part from the beginning and the same long part from the end of the sign were truncated and only the central part was used in the utterance. The results for UWB-06-SLR-A corpus with LBP + Δ^2 features are shown in the following figure 6.27.

The results show that after the truncation of *preparation* and *retraction* part of the sign the recognition accuracy is still good, in the presented graph the accuracy dropped from 94.23%

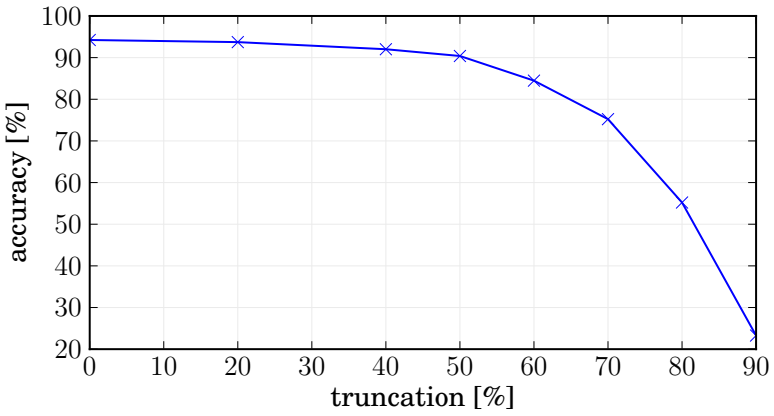


Figure 6.27: Accuracy for continuous sign language recognition on UWB-06-SLR-A corpus with LBP + Δ^2 features for different truncations of the signs. For example, 50% value means that 25% of the sign was truncated from the beginning and the same part from the end.

to 92.01% when 40% of the sign was truncated, or to 90.39% for 50%. With this result, the proposed continuous recognition system was able to recognize whole sentences with signs where the effect of "neutral position" was suppressed.

6.5 Search by Example

The *search by example* problem, described in general in section 5.5, is designed, implemented and evaluated. The task is to perform searching in sign language videos based on the user-given image of one or multiple hand, captured for example through a webcam. The result is a sorted list of the videos containing the requested hand shapes.

Often, such systems used in other fields are based on the *query-by-example* paradigm, where various features are preliminarily extracted from the stored images or videos. The query is an image, a set of images or a video, on which the same features are calculated and stored data are retrieved and ranked with respect to their similarity with the search query.

The task can be formalized as follows. The set of N_V sign language videos, each containing one signer performing a single sign or a longer utterance, is denoted as $\mathbf{V} = [V_1, V_2, \ldots, V_{N_V}]$. The user provides search query as a set of N_Q images $\mathbf{Q} = [Q_1, Q_2, \ldots, Q_{N_Q}]$, each containing one hand. The result is a distance of the query input $\mathbf{Q}$ with indexed video dataset $\mathbf{V}$, denoted as $D = [d_1, d_2, \ldots, d_{N_V}]$, where d_i is the distance between the query and i-th indexed video. The indices of D with the lowest distances indicate the best matching videos for the given query.

There are several design questions. How to extract features from the input images $\mathbf{Q}$, how to preprocess the dataset $\mathbf{V}$ so that the search is fast, and mainly, how to calculate the distance vector D. The proposed method originated from text search engines, and is used in visual search engines [CMPM11] where both the query and all indexed images are represented as a sparse vector of visual word occurrences. Then, the similarity between the query vector and each image vector is calculated. Similar approach is used for example in scene categorization

problem [FFP05]. The whole process can be separated into *indexing* and *searching* phase, or in analogous machine learning terms to *training* and *testing* phase. Both are depicted in figure 6.28 and are discussed in the following section.

6.5.1 Indexing

As is shown in the figure 6.28, the indexing consists of several steps. The first is hand feature extraction. Here we employ tracking and manual component feature extraction methods proposed in previous sections 6.2.2 and 6.2.3. All videos $\mathbf{V}$ are processed, resulting in a set of feature vectors describing the hands present in the video. Denote $\mathbf{V}^F = [V_1^F, V_2^F, \ldots, V_{N_V}^F]$, where V_i^F is a set of feature vectors describing hands in the i-th video.

Three different hand feature descriptors were used and their performance was compared: hRDF, HOG and LBP. Additionally, one combination of LBP+hRDF feature vectors was used.

With the available tracking and selected feature extraction method, the sets of feature vectors $\mathbf{V}^F$ can be computed.

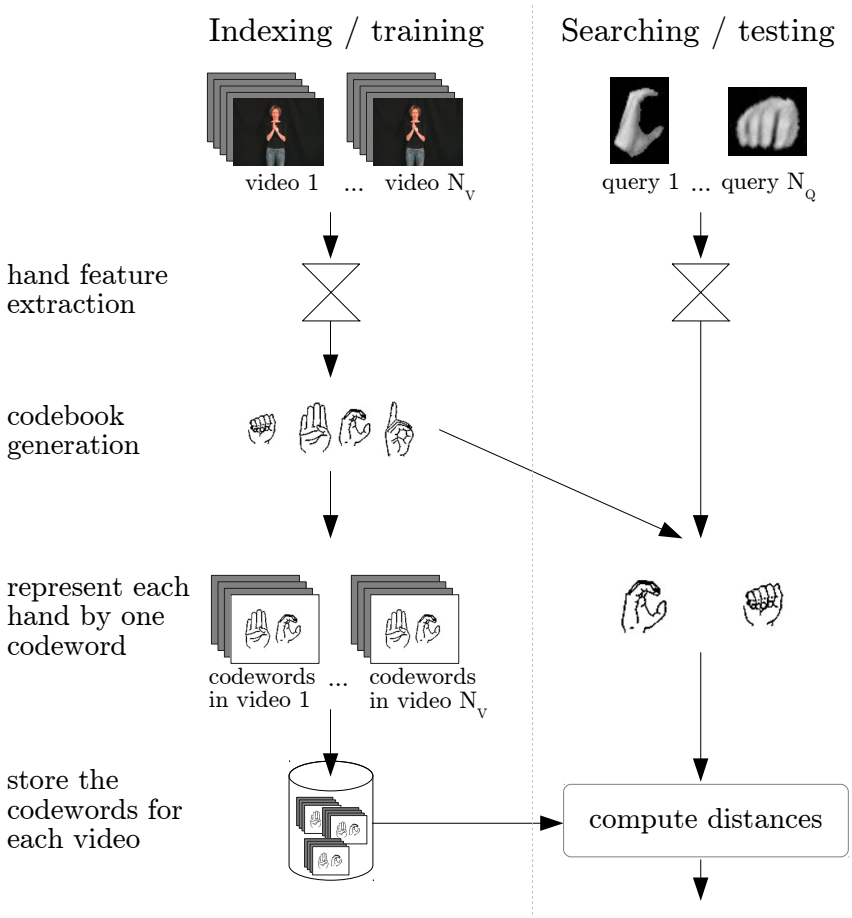


Figure 6.28: Scheme of the proposed *search by example* system, consisting of the indexing and searching subsystem.

The next step is codebook generation, where the feature vectors are converted to "codewords" (the analogy comes from words in text documents), which produces a "codebook" (analogous to a word dictionary). The codewords can be interpreted as representatives of several similar hand images. To generate such codewords, k-means clustering was performed on a subset of 20000 randomly selected feature vectors computed on the *UWB-07-SLR-P hand corpus* images, which contains 257354 hand images, thus 7.77% of the available images were used for the clustering. The number of the clusters k, *codebook size*, was fixed to $k = 2000$. The codewords are defined as the centers of the learned clusters. Now, each image containing a hand can be mapped to a certain codeword. Examples of hand images that are mapped to the same cluster, based on hRDF features, is shown in figure 6.29.

The figure 6.30 shows a two dimensional PCA projection of the feature space and sample images from the clusters at given locations. All three methods, hRDF, HOG and LBP are shown, each of the methods has different similarity results. Similar hand shapes should be depicted near to each other, different hand shapes far.

Finally, the indexation is completed by storing all the mapped codewords for each video $\mathbf{V}^C = [V_1^C, V_2^C, \ldots, V_{N_V}^C]$, computed from the feature vectors $\mathbf{V}^F$. Thus, i-th video is represented by a short vector V_i^C containing indexes of codewords.

To add a new video to the search index, it is not necessary to repeat the whole training process. The codebook can remain the same and only new codewords will be detected in the video and stored.

Additionally, to speed up the search process, a rectangular distance matrix $D = [d_{ij}]$ of dimension k is calculated, where the element d_{ij} is a Euclidean distance between the center of i-th and j-th cluster. This is used whenever a distance between two codewords is needed.

6.5.2 Searching

The search query consists of N_Q images $\mathbf{Q} = [Q_1, Q_2, \ldots, Q_{N_Q}]$, each containing one hand. The images are normalized and the same feature descriptor that was used for indexing is applied to the query images. The codebook maps each feature vector into a codeword, which forms a query represented by the codewords $Q^C = [q_1^C, q_2^C, \ldots, q_{N_Q}^C]$. Now, the goal is to calculate distances between the query codewords Q^C and indexed videos that are represented by sets of codewords of the same codebook $\mathbf{V}^C$.

The distance between i-th query codeword q_i^C and j-th video codeword set V_j^C, denoted as d_{ij}^Q, is computed as a sum of distances between the i-th cluster and M nearest codewords in V_j^C. The selection of M was based on evaluation of the system and showed the best results for $M = 3$, for all used feature descriptors. So, the best matching videos for the given query must

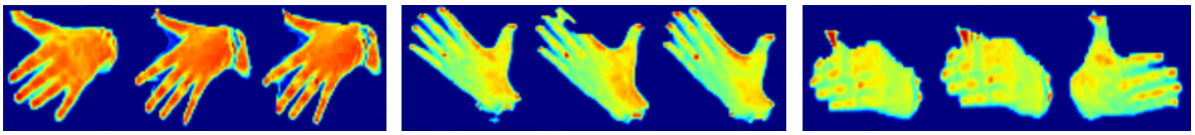


Figure 6.29: Example of 3 clusters resulted from k-means clustering based on hRDF features.

(a) hRDF features **(b)** HOG features **(c)** LBP features

Figure 6.30: Two dimensional PCA projection of the feature space demonstrated on sample images from the clusters.

contain the queried hand shapes or very similar ones at least three times each.

In case the query contains more than one query image, the computed distances d_{ij}^Q must be combined and form a single final distance D_j^Q. A simple sum of all d_{ij}^Q can be used, but better results were obtained when the distances are interpreted as probabilities, multiplied together and then reinterpreted back as a distance:

$$D_j^Q = 1 - \prod_{i=1}^{N_Q}(1 - d_{ij}^Q) \tag{6.1}$$

Each factor $1-d_{ij}^Q$ can be interpreted as a probability that the i-th query codeword is present in j-th video. This probability is not based on measurement or any distribution estimation, only a naive relation is used considering that more distant codewords are less probable than the closer ones. The feature space for all feature descriptors is normalized so that the distance between any real feature vectors lies in interval $< 0, 1 >$, thus the same interval applies for d_{ij}^Q, which imply that the possible values of $1 - d_{ij}^Q$ are kept in $< 0, 1 >$ and allows to interpret it as a probability value.

Finally, when all the distances D_j^Q between the query and j-th video are calculated, the videos can be sorted depending on these distances, thus creating the result of the search.

If the feature descriptors are not rotation invariant, some level of rotation invariance can be added by appending artificial query images generated from the original images by a rotation transformation for several manually selected angles.

6.5.3 Evaluation

The evaluation was measured on UWB-07-SLR-P corpus, where the codebook was trained on UWB-07-SLR-P hand corpus. Four different hand feature descriptors were used: uniform *LBP* with radius 2 computed on normalized image with the size of 64 pixels, with 8-neighborhood; *HOG* with 8 orientations and 8 pixels per cell, computed on normalized image with the size of 112 pixels; *hRDF* with 72 angle bins, computed on normalized image with the size of 100 pixels. *LBP+hRDF* is a fusion of two previously described feature descriptors,

method		position of correct result for **one hand** search					
		median	mean	top 1	top 5	top 10	top 20
LBP+hRDF	0°	**5**	**19.1**	28.5%	50.2%	**62.7%**	**75.5%**
	10°	**13**	39.4	12.3%	30.3%	43.2%	60.8%
	20°	**18**	**49.7**	**8.2%**	**22.4%**	**34.1%**	**53.6%**
HOG	0°	**5**	24.4	**31.3%**	**51.3%**	61.9%	74.0%
	10°	18	48.1	8.8%	24.1%	36.6%	53.5%
	20°	28	73.7	5.1%	15.5%	26.5%	43.5%
hRDF	0°	6	22.0	28.2%	48.4%	61.9%	74.5%
	10°	14	**37.8**	**13.1%**	**30.6%**	**43.4%**	**58.3%**
	20°	23	52.7	5.9%	21.2%	33.4%	47.7%
LBP	0°	9	26.4	22.0%	42.2%	53.9%	68.4%
	10°	19	52.7	8.7%	21.7%	33.8%	52.3%
	20°	28	71.6	6.9%	15.9%	23.6%	39.3%
normalized central moments η_i	0°	11	51.6	18.8%	38.8%	49.8%	64.4%
angle+solidity+extent+ratio	0°	17	63.8	15.1%	32.6%	42.4%	53.4%
Hu moments I_i	0°	26	97.5	10.8%	23.3%	30.9%	44.6%

Table 6.5: Search results - one hand query

where the resulting feature vector is a concatenation of single feature vectors.

The evaluation is based on selection of random search queries, performing the search and measurement of the position in the result, where the sign from which the query images originated appears. Because the corpus contains several repetitions of the same sign, performed by several signers, all these signs are expected to contain the same or similar hand configurations. Thus, the result is considered as correct if the video contains the same sign from which the query images originated. Only a subset of UWB-07-SLR-P corpus was used, containing 220 different both-handed signs.

Several evaluation measurements were used. *Median* and *mean* positions indicate at which position the correct sign appeared in the results. *Top-N* rates show in how many cases the correct result appeared in first N results.

The tables 6.5 and 6.6 show the results based on evaluation of 601 random queries. The first table is for queries containing one hand, the second is for two hand queries.

The search based on one hand queries achieved the best results with *LBP+hRDF* features, with the median position of correct result on 5th position, but increased to 18 when 20°rotation invariance was considered. The results in this table show that all four presented feature descriptors are more suitable for this task than normalized central moments, Hu moments or a combination of scalar region descriptors.

The second table 6.6 with the result for two hand search shows that the performance increased rapidly compared to the one hand case. The best hand descriptor was *LBP+hRDF* when no rotation invariance was considered, *hRDF* otherwise. Despite very promising results

		position of correct result for **two hand** search					
method		median	mean	top 1	top 5	top 10	top 20
LBP+hRDF	0°	0	3.9	**77.1**%	**91.6**%	**94.7**%	**97.1**%
	10°	**2**	9.2	46.0%	70.8%	83.2%	90.9%
	20°	5	15.1	29.9%	54.1%	68.6%	83.7%
HOG	0°	0	3.9	75.3%	86.5%	91.5%	95.2%
	10°	3	13.2	35.9%	60.0%	73.8%	83.1%
	20°	9	28.6	15.3%	36.7%	53.4%	68.3%
hRDF	0°	0	**3.2**	72.8%	87.6%	93.3%	95.7%
	10°	**2**	**6.7**	**48.8**%	**73.2**%	**83.9**%	**91.3**%
	20°	**4**	**13.2**	**30.9**%	**55.5**%	**71.5**%	**84.2**%
LBP	0°	0	5.1	64.2%	79.7%	87.7%	92.3%
	10°	4	18.0	34.1%	55.8%	70.8%	84.0%
	20°	9	34.6	22.9%	40.3%	54.4%	66.9%

Table 6.6: Search results - two hand query

shown in this table, a human-based evaluation is needed. This artificial evaluation has both advantages and disadvantages when compared to the real world usage. Some advantages are flawless images directly retrieved from the corpus or the images coming from the same signers. Disadvantage is for example the random selection of the query images, which can select some common hand image which has no discriminative power; the human will usually try to build a query from some discriminative hand images.

7 | Conclusion and Future Work

In this work, a sign language recognition (SLR) system has been developed and evaluated, both for recognition of isolated signs and continuous utterances. The SLR system uses statistical approach for the recognition tasks. Additionally, a *search by example* system for searching in video data containing sign language utterances using a query consisting of one or multiple hand images has been proposed and evaluated.

Both systems share some approaches and methods, and some are unique depending on the application. They form a basis for experiments so that each system can be evaluated as a whole. The focus was put into feature extraction methods, including robust hand tracking resolving occlusions with face. Both manual (hands) and nonmanual (face) components of signs were studied. Then, sign modeling that employs hidden Markov models and the first step of data-driven construction of phoneme sub-units was designed and evaluated.

The novel *search by example* system based on image-based queries showed promising results for the real world usage, for example in interactive sign language dictionaries.

All the original goals of the thesis were fulfilled:

Corpora preparation Two corpora of isolated sign recordings, UWB-06-SLR-A and UWB-07-SLR-P, were collected. Both were recorded in laboratory conditions, by multiple speakers with multiple repetitions, thus the corpora are directly targeted for SLR experiments.

Automatic sign language recognition system was developed, based on hidden Markov models which are widely used in speech recognition systems. The system is capable to recognize both isolated and continuous utterances. The evaluation of the system was performed on various features and their combinations on the UWB-06-SLR-A and UWB-07-SLR-P corpora. The best achieved recognition accuracies of isolated signs were 98.65% / 95.06% (signer dependent / independent recognition) for the UWB-06-SLR-A corpora, which consist of smaller number of signs but was recorded by higher number of signers than UWB-07-SLR-P, where the best accuracies are 93.19% / 34.47%. The evaluation of continuous utterances was performed on artificially generated utterances, with 97.19% / 86.89% accuracy for UWB-06-SLR-A and 86.00% / 19.79% for UWB-07-SLR-P dataset. A deeper inspection on the results and a discussion why the results decreased for signer independent recognition on UWB-07-SLR-P dataset was discussed in section 6.3.3.

Feature extraction Since special recording devices were avoided, the only considered sources of data are digital cameras that provide series of image frames. Several existing and new

approaches for feature extraction were examined and their performance in the classification tasks was compared. Both low level image features and higher level appearance-based features incorporating hand and head tracking were studied. Best recognition results in particular tasks were achieved using local binary patterns (LBP), their fusion with hand coordinates, and hRDF (hand shape radial distance function) shape descriptor that was proposed in this work as an extension to earlier presented RDF shape descriptor.

Hand tracking and occlusion handling Because the hand position, orientation and its configuration are crucial features for recognition, an algorithm that separates head and hand pixels was employed to ease further hand tracking and feature extraction. The algorithm allows to interpolate face region during the occlusion with hand, and enables the use of face extraction methods that fail in case of occlusion. The algorithm is an extension of a recently published method.

Sub-units An iterative algorithm that serves as a first step for data-driven construction of sub-units was proposed, employing Gaussian mixture models (GMM). Although the sub-units were not identified from the dataset, the algorithm created HMM models based on shared observation probabilities among all the signs. Although the recognition accuracy for such a HMM models was lower than with the independent models, the results showed promising outcomes and the possibility that the sub-units can be constructed employing clustering on the HMM states that were identified by the proposed algorithm. During the course of the work, systems based on similar principles were introduced by other researchers [Kel10] or [TPM10], indicating high activity in this particular subfield of SLR.

Feature selection and fusion The principal component analysis (PCA) method was applied on the considered features, and its influence for the recognition was studied. Additionally, feature fusion performed at feature level was studied, investigating different combinations of features, and showed better results in some recognition tasks.

Search by example Beyond the SLR system, searching in video corpora by user given example was studied. The example consists of one or multiple images. The system searches for the parts in videos that contain such hand configurations as given by the user. The system was built and evaluated on the larger UWB-07-SLR-P corpus. When using one image query, the median position of correct result was 5, which is acceptable for a user. If some level of rotation invariance is considered for real world usage, the median position decreased to 18. In such a case, at least two query images should be provided. Thus, the median position is back at acceptable value 4. The conclusion is that two images containing a hand, used as a search query, provide good search results.

Czech Sign Language online dictionary Complementary to the goals of the thesis, we developed an online sign language dictionary with some unique functionalities. The dictionary supports SignWriting and HamNoSys notations, even for search purposes; a 3D avatar that uses sign language synthesis system as an alternative to videos for sign presentations. This online dictionary can serve as a source of additional data, both video and linguistic, and as a platform where the results of this thesis can be applied into a real-world system. [1].

7.1 Future Work

The field of automatic sign language recognition is growing rapidly. This thesis has covered only a few of the problems of this difficult field. The subtasks that were employed in this work pose a good starting point for further research. Some interesting directions for future work are discussed in the following.

The first direction is to utilize latest corpora, with continuous sign language utterances, recorded with the purpose for SLR experiments. This would allow investigating effects of grammar and coarticulation, where each sign is influenced by neighboring signs in the sentence.

The work on sign modeling can be extended by some modifications of HMMs, that allow some level of asynchrony for standalone streams of hands and head features, such as *product HMM* (as was discussed in section 5.2.3).

The feature extraction subtask can examine other existing feature descriptors, or create new ones that are suitable for robust hand and face feature extraction.

As the hand tracking and occlusion handling is already performing well, some experiments can be done in more unconstrained environments, for instance with multiple signers, under different lighting conditions etc. Other currently examined tracking methods use new input devices that provide both RGB image and 3D depth map, thus the feature extraction can be more robust.

Probably the largest potential for advances lies in sub-unit identification and modeling, as was discussed in section 6.3.4. The challenge is to identify the sub-units not only by data-driven methods, but with cooperation with linguists.

Other big potential lies in creation of language models that can improve recognition accuracy for continuous SLR with large vocabularies. The main problem here is the data collection and annotation.

[1] Czech Sign Language online dictionary is available in English at `http://signs.zcu.cz`, current production version in Czech language is available at `http://znaky.zcu.cz`

Bibliography

[AAA+08] Oya Aran, Ismail Ari, Lale Akarun, Erinc Dikici, Siddika Parlak, Murat Saraclar, Pavel Campr, and Marek Hruz. Speech and Sliding Text Aided Sign Retrieval from Hearing Impaired Sign News Videos. *Journal on Multimodal User Interfaces*, 2(2):117–131, June 2008.

[AAA+09] Oya Aran, Ismail Ari, Lale Akarun, Bülent Sankur, Alexandre Benoit, Alice Caplier, Pavel Campr, Ana Huerta Carrillo, and Francois-Xavier Fanard. SignTutor: An Interactive System for Sign Language Tutoring. *IEEE Multimedia*, 16(1):81–93, January 2009.

[ABCA09] Oya Aran, Thomas Burger, Alice Caplier, and Lale Akarun. A Belief-based Sequential Fusion Approach for Fusing Manual Signs and Non-manual Signals. *Pattern Recognition*, 42(5):812–822, May 2009.

[AHEK10] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal Fusion for Multimedia Analysis: A Survey. *Multimedia Systems*, 16(6):345–379, April 2010.

[AK07] Ulrich Von Agris and Karl-Friedrich Kraiss. Towards a Video Corpus for Signer-Independent Continuous Sign Language Recognition. In *GW 2007 The 7th International Workshop on Gesture in Human-Computer Interaction and Simulation*, pages 10–11, Lisbon, Portugal, 2007.

[ANS+08] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, and Ashwin Thangali. The American Sign Language Lexicon Video Dataset. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, number June, pages 1–8. IEEE, June 2008.

[ANS+10] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Ashwin Thangali, Haijing Wang, and Quan Yuan. Large Lexicon Project : American Sign Language Video Corpus and Sign Language Indexing / Retrieval Algorithms. In *Workshop on the Representation and Proceedings of Sign Languages: Corpora and Sign Language Technologies (CSLT)*, pages 11–14, 2010.

[Ara08] Oya Aran. *Vision Based Sign Language Recognition: Modeling and Recognizing Isolated Signs With Manual and Non-manual Components.* PhD thesis, Bogazici University, 2008.

[Aus] Auslan Signbank. http://www.auslan.org.au.

[Bak75] J. Baker. The DRAGON System - An Overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):24–29, February 1975.

[BB12] Annelies Braffort and Lela Boutora. DEGELS1: A Comparable Corpus of French Sign Language and Co-speech Gestures. In *International Conference on Language Resources and Evaluation*, pages 2426–2429, 2012.

[BEHZ08] P. Buehler, M. Everingham, D. Huttenlocher, and A. Zisserman. Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts. In *Procedings of the British Machine Vision Conference 2008*, pages 110.1–110.10. British Machine Vision Association, 2008.

[BEZ10] Patrick Buehler, Mark Everingham, and Andrew Zisserman. Employing Signed TV Broadcasts for Automated Learning of British Sign Language. In *In proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Techhnologies*, 2010.

[BM01] Simon Baker and Iain Matthews. Equivalence and Efficiency of Image Alignment Algorithms. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–1090–I–1097. IEEE Comput. Soc, 2001.

[BM04] Simon Baker and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, February 2004.

[BN04] M. Bisani and H. Ney. Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–409–12. IEEE, 2004.

[BOP97] M. Brand, N. Oliver, and A. Pentland. Coupled Hidden Markov Models for Complex Action Recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999. IEEE Comput. Soc, 1997.

[BSD+06] J Bungeroth, D Stein, Philippe Dreuw, M Zahedi, and H Ney. A German Sign Language Corpus of the Domain Weather Report. In *Fifth International Conference on Language Resources and Evaluation*, pages 2000–2003, Genoa, Italy, 2006.

[BSD+08] Jan Bungeroth, Daniel Stein, Philippe Dreuw, Hermann Ney, Sara Morrissey, Andy Way, and Lynette Van Zijl. The ATIS Sign Language Corpus. In *LREC*, Marrakech, Morocco, 2008.

[BSS01] Penny Boyes Braem and Rachel Sutton-Spence. *The Hands Are the Head of the Mouth.* Signum-Verl., Hamburg, 2001.

[Bur98] Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[BWK+04] R Bowden, D Windridge, T Kadir, A Zisserman, and M Brady. A Linguistic Feature Vector for the Visual Interpretation of Sign Language, 2004.

[Cí06] Petr Císař. *Využití metod odezírání ze rtů pro podporu rozpoznávání řeči.* PhD thesis, University of West Bohemia in Pilsen, 2006.

[CB09] Helen Cooper and Richard Bowden. Sign Language Recognition : Working with Limited Corpora. In *Proceedings of the International Conference on Universal Access in Human-Computer Interaction. Addressing Diversity*, pages 472–481, San Diego, CA, USA, 2009.

[CB10] Helen Cooper and Richard Bowden. Sign Language Recognition using Linguistically Derived Sub-Units. In *Proceedings of the Language Resources and Evaluation Conference Workshop on the Representation and Processing of Sign Languages : Corpora and Sign Languages Technologies*, volume 1, pages 17–23, 2010.

[CHL+10] Pavel Campr, Marek Hrúz, Jiří Langer, Jakub Kanis, Miloš Železný, and Luděk Müller. Towards Czech On-line Sign Language Dictionary - Technological Overview and Data Collection. In *LREC 2010, Seventh international conference on language resources and evaluation; 4th workshop on the representation and processing of sign languages: corpora and sign language technologies*, pages 41–44, Valletta, Malta, 2010.

[CHT08] Pavel Campr, Marek Hrúz, and Jana Trojanová. Collection and Preprocessing of Czech Sign Language Corpus for Sign Language Recognition. In ELRA, editor, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008.

[CHv07] Pavel Campr, Marek Hrúz, and Miloš Železný. Design and Recording of Signed Czech Language Corpus for Automatic Sign Language Recognition. *Interspeech 2007*, pages 678–681, 2007.

[CKK+12] Pavel Campr, Jakub Kanis, Zdeněk Krňoul, Marek Hrúz, Luděk Müller, and Železný Miloš. Czech Sign Language Online Dictionary. http://signs.zcu.cz, 2012.

[CKM07] Chi-Ho Chan, Josef Kittler, and Kieron Messer. Multi-scale Local Binary Pattern Histograms for Face Recognition. In Seong-Whan Lee and Stan Li, editors, *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 809–818. Springer Berlin / Heidelberg, 2007.

[CMPM11] Ondrej Chum, Andrej Mikulik, Michal Perdoch, and Jiri Matas. Total Recall II: Query Expansion Revisited. In *CVPR 2011*, pages 889–896. IEEE, June 2011.

[CTCG95] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[CZ08] Onno Crasborn and Inge Zwitserlood. The Corpus NGT: An Online Corpus for Professionals and Laymen. In *3rd Workshop on the Representation and Processing of Sign Languages (LREC) ; Construction and Exploitation of Sign Language Corpora*, pages 44–49, Marrakech, Morocco, 2008. ELDA.

[DFN10] Philippe Dreuw, Jens Forster, and Hermann Ney. Tracking Benchmark Databases for Video-Based Sign Language Recognition. In *ECCV International Workshop on Sign, Gesture, and Activity*, 2010.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, 2001.

[DNA+08] Philippe Dreuw, Carol Neidle, Vassilis Athitsos, Stan Sclaroff, and Hermann Ney. Benchmark Databases for Video-Based Automatic Sign Language Recognition. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1115–1120, Marrakech, Morocco, 2008. European Language Resources Association (ELRA).

[DRD+07] Philippe Dreuw, D Rybach, T Deselaers, M Zahedi, and H Ney. Speech Recognition Techniques for a Sign Language Recognition System. In *Interspeech 2007*, pages 2513–2516, Antwerp, Belgium, 2007.

[Dre12] Philippe Dreuw. *Probabilistic Sequence Models for Image Sequence Processing and Recognition*. PhD thesis, RWTH Aachen University, 2012.

[DT05] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[ECG+11] Ralph Elliott, Helen Cooper, John Glauert, Richard Bowden, and Francois Lefebvre-Albaret. Search-By-Example in Multilingual Sign Language Databases. In *Proceedings of the Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, Dundee, Scotland, 2011.

[Ech] ECHO Database. http://www.let.ru.nl/sign-lang/echo.

[EFH+12] Eleni Efthimiou, Stavroula-Evita Fotinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, P. Maragos, and F. Lefebvre-Albaret. Sign Language technologies and resources of the Dicta-Sign project. In *Proc. of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon. Satellite Workshop to the eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 37–44, Istanbul, Turkey, 2012.

[ETC98] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting Face Images Using Active Appearance Models. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 300–305. IEEE Comput. Soc, 1998.

[FFP05] Li Fei-Fei and Pietro Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005.

[FGGC04] Gaolin Fang, Xiujuan Gao, Wen Gao, and Yiqiang Chen. A Novel Approach to Automatically Extracting Basic Units from Chinese Sign Language. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, pages 454–457 Vol.4, 2004.

[Fod02] Imola Fodor. A Survey of Dimension Reduction Techniques. Technical report, Lawrence Livermore National Laboratory, Livermore, CA, USA, 2002.

[FR94] William T. Freeman and Michal Roth. Orientation Histograms for Hand Gesture Recognition. In *In International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1994.

[FSH+12] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus Piater, and Hermann Ney. RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus. In *International Conference on Language Resources and Evaluation*, pages 3785–3789, Istanbul, Turkey, 2012.

[GC11] Matilde Gonzalez and Christophe Collet. Robust Body Parts Tracking Using Particle Filter and Dynamic Template. In *2011 18th IEEE International Conference on Image Processing*, pages 529–532. IEEE, September 2011.

[GCD10] Matilde Gonzalez, Christophe Collet, and R. Dubot. Head Tracking and Hand Segmentation during Hand over Face Occlusion in Sign Language. In *International Workshop on Sign, Gesture, and Activity (SGA) in Conjunction with ECCV 2010*, 2010.

[GH06] Paul Goh and Eun-jung Holden. Dynamic Fingerspelling Recognition using Geometric and Motion Features. In *2006 International Conference on Image Processing*, pages 2741–2744. IEEE, 2006.

[GJ97] Zoubin Ghahramani and Michael I. Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29(2-3):245–273, 1997.

[GMW97] D. Gibbon, R. Moore, and R. Winski. *Handbook of Standards and Resources for Spoken Language Systems*. Mouton De Gruyter, 1997.

[HAS09] Junwei Han, George Awad, and Alistair Sutherland. Modelling and Segmenting Subunits for Sign Language Recognition Based on Hand Motion Analysis. *Pattern Recognition Letters*, 30(6):623–633, April 2009.

[HCD+11] Marek Hrúz, Pavel Campr, Erinç Dikici, Ahmet Alp Kındıroğlu, Zdeněk Krňoul, Alexander Ronzhin, Haşim Sak, Daniel Schorno, Hülya Yalçın, Lale Akarun, Oya Aran, Alexey Karpov, Murat Saraçlar, and Milos Železný. Automatic Fingersign-to-speech Translation System. *Journal on Multimodal User Interfaces*, 4(2):61–79, 2011.

[HKCM11] Marek Hrúz, Zdeněk Krňoul, Pavel Campr, and Luděk Müller. Towards Automatic Annotation of Sign Language Dictionary Corpora. In *Lecture Notes in Computer Science. Text, Speech and Dialogue*, pages 331–339. Springer-Verlag, 2011.

[HLM04] N. Habili, C.C. Lim, and a. Moini. Segmentation of the Face and Hands in Sign Language Video Sequences Using Color and Motion Cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(8):1086–1097, August 2004.

[HLO05] Eun-Jung Holden, Gareth Lee, and Robyn Owens. Australian Sign Language Recognition. *Machine Vision and Applications*, 16(5):312–320, November 2005.

[Hol93] Judith A. Holt. Stanford Achievement Test—8th Edition: Reading Comprehension Subgroup Results. In *American Annals of the Deaf 138(2)*, pages 172–175, 1993.

[HRH07] G A Holt, M J T Reinders, and E A Hendriks. Multi-Dimensional Dynamic Time Warping for Gesture Recognition. *Thirteenth annual conference of the Advanced School for Computing and Imaging*, 2007.

[Hru09] J. Hrubý. Tak kolik těch sluchově postižených u nás vlastně je? *Speciální pedagogika*, 19(4), 2009.

[HSS02] Yasushi Hamada, Nobutaka Shimada, and Yoshiaki Shirai. Hand Shape Estimation Using Sequence of Multi-Ocular Images Based on Transition Network. *VI02*, page 362, 2002.

[HTH00] P Hong, M Turk, and T Huang. Gesture Modeling and Recognition using Finite State Machines. citeseer.ist.psu.edu/hong00gesture.html, 2000.

[HTv11] Marek Hrúz, Jana Trojanová, and Miloš Železný. Local Binary Pattern Based Features for Sign Language Recognition. *Pattern Recognition and Image Analysis*, 21(3):398–401, 2011.

[IB98] Michael Isard and Andrew Blake. CONDENSATION - Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[JBM75] F. Jelinek, L. Bahl, and R. Mercer. Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech. *IEEE Transactions on Information Theory*, 21(3):250–256, May 1975.

[JGY+09] Feng Jiang, Wen Gao, Hongxun Yao, Debin Zhao, and Xilin Chen. Synthetic Data Generation Technique in Signer-Independent Sign Language Recognition. *Pattern Recognition Letters*, 30(5):513–524, April 2009.

[Kel10] Daniel Kelly. *Computational Models for the Automatic Learning and Recognition of Irish Sign Language*. PhD thesis, National University of Ireland Maynooth, 2010.

[KHDM98] Josef Kittler, Mohamad Hatef, RPW Duin, and J. Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[KJV+12] Matti Karppa, Tommi Jantunen, Ville Viitaniemi, Jorma Laaksonen, Birgitta Burger, and Danny De Weerdt. Comparing Computer Vision Analysis of Signed Language Video with Motion Capture Recordings. In *International Conference on Language Resources and Evaluation*, pages 2421–2425, 2012.

[KKC+11] Zdeněk Krňoul, Jakub Kanis, Pavel Campr, Miloš Železný, and Luděk Müller. Sign Speech Synthesis System. In *International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, Berlin, Germany, 2011.

[KMM11] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-Learning-Detection. *IEEE transactions on pattern analysis and machine intelligence*, 6(1):1–14, December 2011.

[KPS03] J. Kovac, P. Peer, and F. Solina. Human Skin Color Clustering for Face Detection. In *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, volume 2, pages 144–148. IEEE, 2003.

[Kuc05] Lucie Kuchařová. Jazyk neslyšících, 2005.

[KYA+11] A A Kindiroglu, H Yalcin, O Aran, M Hruz, P Campr, L Akarun, and A Karpov. Multi-lingual Fingerspelling Recognition for Handicapped Kiosk. *Pattern Recognition and Image Analysis*, 21(3):402–406, September 2011.

[KYSD06] E. Konukoglu, E. Yoruk, B. Sankur, and J. Darbon. Shapebased Hand Recognition. *IEEE Trans. Image Processing*, 15(7):1803–1815, 2006.

[KZJM06] Jakub Kanis, Jirí Zahradil, Filip Jurčíček, and Luděk Müller. Czech-Sign Speech Corpus for Semantic Based Machine Translation. In Petr Sojka, Ivan Kopecek, and Karel Pala, editors, *Proceedings of TSD 2006*, volume 4188 of *Lecture Notes in Computer Science*, pages 613–620. Springer, 2006.

[LE09] Stephan Liwicki and Mark Everingham. Automatic Recognition of Fingerspelled Words in British Sign Language. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, number iv, pages 50–57. IEEE, June 2009.

[Lev66] Vladimir I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[Lid77] Scott K. Liddell. *An Investigation into the Syntactic Structure of American Sign Language*. PhD thesis, University of California, San Diego, 1977.

[Lid03] Scott K. Liddell. *Grammar, Gesture, and Meaning in American Sign Language*. Cambridge University Press, Cambridge, 2003.

[LM02] R. Lienhart and J. Maydt. An Extended set of Haar-like Features for Rapid Object Detection. In *Proceedings. International Conference on Image Processing*, volume 1, pages I–900–I–903. IEEE, 2002.

[Low99] D.G. Lowe. Object Recognition from Local Scale-invariant Features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2. IEEE, 1999.

[LTC97] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic Interpretation and Coding of Face Images using Flexible Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

[MB04] Iain Matthews and Simon Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, November 2004.

[MB11] S. Moore and R. Bowden. Local Binary Patterns for Multi-view Facial Expression Recognition. *Computer Vision and Image Understanding*, 115(4):541–558, April 2011.

[Mil] MILK: Machine Learning Toolkit. http://packages.python.org/milk.

[MLY+12] Dimitris Metaxas, Bo Liu, Fei Yang, Peng Yang, Nicholas Michael, and Carol Neidle. Recognition of Nonmanual Markers in American Sign Language (ASL) Using Non-Parametric Adaptive 2D-3D Face Tracking. In *International Conference on Language Resources and Evaluation*, pages 2414–2420, 2012.

[OKA12] I. Oikonomidis, N. Kyriazis, and a. a. Argyros. Tracking the Articulated Motion of Two Strongly Interacting Hands. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1862–1869, June 2012.

[ope12] OpenCV. http://opencv.org/, 2012.

[Pea01] Karl Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine Series 6*, 2(11):559–572, November 1901.

[PMMR06] J. Psutka, L. Müller, J. Matoušek, and V. Radová. *Mluvíme s počítačem česky*. Academia, Prague, 2006.

[PNJS05] Pedro R. Peres-Neto, Donald a. Jackson, and Keith M. Somers. How Many Principal Components? Stopping Rules for Determining the Number of Non-trivial Axes Revisited. *Computational Statistics & Data Analysis*, 49(4):974–997, June 2005.

[PSH97] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):677–695, 1997.

[RJ86] L. Rabiner and B. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, June 1986.

[RS83] Levinson S.E. Rabiner and M.M. Sondhi. On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition. *The Bell System Technical Journal*, 62(4), 1983.

[RTPM10] Anastasios Roussos, Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. Affine-invariant Modeling of Shape-appearance Images Applied on Sign Language Handshape Classification. In *2010 IEEE International Conference on Image Processing*, pages 1417–1420. IEEE, September 2010.

[RVCB03] Aditya Ramamoorthy, Namrata Vaswani, Santanu Chaudhury, and Subhashis Banerjee. Recognition of Dynamic Hand Gestures. *Pattern Recognition*, 36(9):2069–2081, September 2003.

[SASA09] Pinar Santemiz, Oya Aran, Murat Saraclar, and Lale Akarun. Automatic Sign Segmentation from Continuous Signing via Multiple Sequence Alignment. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2001–2008, September 2009.

[SB08] Leonid Sigal and Michael J Black. Combined Discriminative and Generative Articulated Pose and Non-rigid Shape Estimation. *Advances in Neural Information Processing Systems, (NIPS 2007)*, 20:1337–1344, 2008.

[SdS07] Paul Smith, Niels da Vitoria Lobo, and Mubarak Shah. Resolving Hand over Face Occlusion. *Image and Vision Computing*, 25(9):1432–1448, September 2007.

[SHB07] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. CL Engineering, 3rd editio edition, 2007.

[SK87] L. Sirovich and M. Kirby. Low-dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America A*, 4(3):519, March 1987.

[SLM06] Wendy Sandler and Diane Lillo-Martin. *Sign Language and Linguistic Universals*. Cambridge University Press, Cambridge, 2006.

[Sto60] William C. Stokoe. *Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf*. University of Buffalo, Buffalo, studies in edition, 1960.

[Ten10] Anna Gineke Ten Holt. *Automatic Sign Language Recognition Inspired by Human Sign Perception*. PhD thesis, Technische Universiteit Delft, 2010.

[THZ+08] Jan Trmal, Marek Hrúz, Jan Zelinka, Pavel Campr, and Luděk Müller. Feature Space Transforms for Czech Sign-Language Recognition. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2008)*, pages 2036–2039. Causal Production Pty ltd., 2008.

[TKM09] Stavros Theodorakis, Athanassios Katsamanis, and Petros Maragos. Product-HMMs for Automatic Sign Language Recognition. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1601–1604. IEEE, April 2009.

[TP91] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, January 1991.

[TPM10] Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. Model-level Data-driven Sub-units for Signs in Videos of Continuous Sign Language. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2262–2265. IEEE, 2010.

[Trm12] Jan Trmal. *Spatio-temporal Structure of Feature Vectors in Neural Network Adaptation*. PhD thesis, University of West Bohemia, Pilsen, 2012.

[vAKK08] Ulrich von Agris, Moritz Knorr, and Karl-Friedrich Kraiss. The Significance of Facial Features for Automatic Sign Language Recognition. *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–6, September 2008.

[Vam96] Peter Wray Vamplew. *Recognition of Sign Language Using Neural Networks*. PhD thesis, University of Tasmania, 1996.

[VC99] T. Vaich and A. Cohen. Comparison of Continuous-Density and Semi-Continuous HMM in Isolated Words Recognition Systems. In *Sixth European Conference on Speech Communication and Technology, EUROSPEECH*, Budapest, Hungary, 1999.

[vHSV11] Jan Švec, Jan Hoidekr, Daniel Soutner, and Jan Vavruška. Web Text Data Mining for Building Large Scale Language Modelling Corpus. *Text, Speech and Dialogue*, 6836:356–363, 2011.

[VJ01] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–511–I–518. IEEE Comput. Soc, 2001.

[VM99] Christian Vogler and Dimitris Metaxas. Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes. *Lecture Notes in Computer Science*, 1739, 1999.

[WCG06] C L Wang, X Chen, and W Gao. Expanding Training Set for Chinese Sign Language Recognition. In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, pages 323–328, Washington, DC, USA, 2006. IEEE Computer Society.

[WHY09] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. An HOG-LBP Human Detector with Partial Occlusion Handling. In *2009 IEEE 12th International Conference on Computer Vision*, pages 32–39. IEEE, September 2009.

[wik12] Wikipedia - The Free Encyclopedia. `http://www.wikipedia.org`, 2012.

[WSM+10] Haijing Wang, Alexandra Stefan, Sajjad Moradi, Vassilis Athitsos, Carol Neidle, and Farhad Kamangar. A System for Large Vocabulary Sign Search. In *Workshop on Sign, Gesture and Activity (SGA)*, 2010.

[YEG+06] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Xunying Andrew Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. The HTK Book, 2006.

[YRT89] S.J. Young, N.H. Russell, and J.H.S Thornton. Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems. Technical report, Cambridge University, 1989.

[Zah07] Morteza Zahedi. *Robust Appearance-based Sign Language Recognition*. PhD thesis, RWTH Aachen University, 2007.

[ZBS+11] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American Sign Language Recognition with the Kinect. *Proceedings of the 13th international conference on multimodal interfaces - ICMI '11*, page 279, 2011.

[ZDR+06] Morteza Zahedi, Philippe Dreuw, David Rybach, Thomas Deselaers, Jan Bungeroth, and Hermann Ney. Continuous Sign Language Recognition – Approaches from Speech Recognition and Available Data Resources. In *Second Workshop on the Representation and Processing of Sign Languages: Lexicographic Matters and Didactic Scenarios*, pages 21–24, Genoa, Italy, 2006.

Authored and Co-authored Works

[1] Jakub Kanis, Petr Peňáz, Pavel Campr, and Marek Hrúz. **A Methodology for Automatic Sign Language Dictionary Creation**. *Universal Learning Design*, Brno, Czech Republic. 2011.

[2] Marek Hrúz, Pavel Campr, Erinç Dikici, Ahmet Alp Kındıroğlu, Zdeněk Krňoul, Alexander Ronzhin, Haşim Sak, Daniel Schorno, Hülya Yalçın, Lale Akarun, Oya Aran, Alexey Karpov, Murat Saraçlar, and Milos Železný. **Automatic Fingersign-to-speech Translation System**. *Journal on Multimodal User Interfaces*. 2011.

[3] Jakub Kanis, Marek Hrúz, and Pavel Campr. **Metodika pro automatizovanou tvorbu slovníku znakového jazyka**. *INSPO*, 2011.

[4] A A Kindiroglu, H Yalcin, O Aran, M Hruz, P Campr, L Akarun, and A Karpov. **Multilingual Fingerspelling Recognition for Handicapped Kiosk**. *Pattern Recognition and Image Analysis*. 2011.

[5] Marek Hrúz, Pavel Campr, Zdenek Krňoul, Milos Železný, Oya Aran, and Pinar Santemiz. **Multi-modal Dialogue System with Sign Language Capabilities**. *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '11*, New York, USA. ACM Press. 2011.

[6] Zdeněk Krňoul, Jakub Kanis, Pavel Campr, Miloš Železný, and Luděk Müller. **Sign Speech Synthesis System**. *International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, Berlin, Germany. 2011.

[7] Marek Hrúz, Zdeněk Krňoul, Pavel Campr, and Luděk Müller. **Towards Automatic Annotation of Sign Language Dictionary Corpora**. *Lecture Notes in Computer Science. Text, Speech and Dialogue*, volume 6836 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin-Heidelberg, Germany. 2011.

[8] Jindřich Matoušek, Zdeněk Hanzlíček, Michal Campr, Zdeněk Krňoul, Pavel Campr, and Martin Grůber. **Web-based System for Automatic Reading of Technical Documents for Vision Impaired Students**. *Lecture Notes in Computer Science. Text, Speech and Dialogue*, volume 6836 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin-Heidelberg, Germany. 2011.

[9] Pavel Campr, Erinç Dikici, Marek Hrúz, Alp Kindiroglu, Zdeněk Krňoul, Alexander Ronzhin, Hasim Sak, Daniel Schorno, Lale Akarun, Oya Aran, Alexey Karpov, Murat Saraclar, and Miloš Železný. **Automatic Fingersign to Speech Translator**. *eNTERFACE'10 The Summer Workshop on Multimodal Interfaces*, 2010.

[10] Zdeněk Krňoul, Marek Hrúz, and Pavel Campr. **Correlation Analysis of Facial Features and Sign Gestures**. *IEEE 10th International conference on signal processing proceedings*, Beijing. 2010.

[11] Pavel Campr, Marek Hrúz, Jiří Langer, Jakub Kanis, Miloš Železný, and Luděk Müller. **Towards Czech On-line Sign Language Dictionary - Technological Overview and Data Collection**. *LREC 2010, Seventh international conference on language resources and evaluation; 4th workshop on the representation and processing of sign languages: corpora and sign language technologies*, Valletta, Malta. 2010.

[12] Marek Hrúz, Pavel Campr, Alexey Karpov, Pinar Santemiz, Oya Aran, and Miloš Železný. **Input and Output Modalities Used in a Sign-language-enabled Information Kiosk**. *Proceedings of SPECOM'2009.* 2009.

[13] Pavel Campr, Marek Hrúz, Alexey Karpov, Pinar Santemiz, Miloš Železný, and Oya Aran. **Sign-language-enabled Information Kiosk**. *eNTERFACE'08 The Summer Workshop on Multimodal Interfaces.* 2009.

[14] Oya Aran, Ismail Ari, Lale Akarun, Bülent Sankur, Alexandre Benoit, Alice Caplier, Pavel Campr, Ana Huerta Carrillo, and Francois-Xavier Fanard. **SignTutor: An Interactive System for Sign Language Tutoring**. *IEEE Multimedia.* 2009.

[15] Pavel Campr, Marek Hrúz, and Jana Trojanová. **Collection and Preprocessing of Czech Sign Language Corpus for Sign Language Recognition**. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. 2008.

[16] Jana Trojanová, Marek Hrúz, Pavel Campr, and Milos Železný. **Design and Recording of Czech Audio-Visual Database with Impaired Conditions for Continuous Speech Recognition**. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).* 2008.

[17] Jan Trmal, Marek Hrúz, Jan Zelinka, Pavel Campr, and Luděk Müller. **Feature Space Transforms for Czech Sign-Language Recognition**. *Proceedings of the 9th Annual Conference of the International Speech Communication Association (Interspeech 2008).* Causal Production Pty ltd. 2008.

[18] Marek Hrúz, Pavel Campr, and Miloš Železný. **Semi-automatic Annotation of Sign Language Corpora**. *LREC 3rd Workshop on the Representation and Processing of Sign Languages Construction and Exploitation of Sign Language Corpora*, Marrakech, Morocco. 2008.

[19] Oya Aran, Ismail Ari, Lale Akarun, Erinc Dikici, Siddika Parlak, Murat Saraclar, Pavel Campr, and Marek Hruz. **Speech and Sliding Text Aided Sign Retrieval from Hearing Impaired Sign News Videos**. *Journal on Multimodal User Interfaces.* 2008.

[20] Marek Hrúz and Pavel Campr. **An Overview of Features for a Sign Language Recognition System from the Database UWB-06-SLR-A**. *The 1st Young Researchers Conference on Applied Sciences (YRCAS 2007)*, Pilsen, Czech Republic. University of West Bohemia. 2007.

[21] Pavel Campr, Marek Hrúz, and Miloš Železný. **Design and Recording of Signed Czech Language Corpus for Automatic Sign Language Recognition**. *Interspeech*. 2007.

[22] Miloš Železný, Pavel Campr, Zdeněk Krňoul, and Marek Hrúz. **Design of a Multi-modal Information Kiosk for Aurally Handicapped People**. *SPECOM 2007 proceedings*. 2007.

[23] Oya Aran, Ismail Ari, Pavel Campr, Erinç Dikici, Marek Hrúz, Deniz Kahramaner, Siddika Parlak, Lale Akarun, and Murat Saraclar. **Speech and Sliding Text Aided Sign Retrieval from Hearing Impaired Sign News Videos**. *eNTERFACE'07 The Summer Workshop on Multimodal Interfaces*, Louvain-la-Neuve. TELE, Universite catholique de Louvain. 2007.

A Appendix Sign Language Recognition of Isolated Signs: Extended Results

The table A.1 shows recognition accuracies for different HMM configurations. The experiment uses LBP + Δ^2 features and UWB-06-SLR-A corpus. The table is presented as an example how the number of HMM states and number of GMM mixture components affects the recognition results.

HMM states	GMM components													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	68.99	85.17	85.62	91.91	92.36	92.58	93.93	93.71	93.71	93.03	94.38	94.38	94.61	94.83
4	77.53	86.74	91.24	93.03	94.61	94.83	95.28	95.06	95.06	95.51	95.73	95.51	95.96	95.51
5	83.60	92.13	93.26	94.61	95.06	95.73	95.96	96.18	96.63	96.18	96.40	95.96	96.18	96.40
6	87.87	93.48	95.73	95.51	94.83	96.40	96.18	95.96	95.96	96.63	96.85	96.85	96.85	97.30
7	89.21	94.38	95.96	95.96	96.85	96.85	97.08	96.63	96.63	96.85	96.85	97.08	97.30	97.30
8	90.34	95.96	96.18	97.30	96.18	96.63	96.63	97.08	96.85	97.08	96.85	97.53	98.20	97.98
9	90.56	95.06	95.28	96.18	96.85	96.85	97.30	96.63	96.85	96.63	96.85	97.08	97.08	97.30
10	90.34	95.51	95.73	95.73	95.73	97.30	97.08	97.30	98.43	97.75	97.98	97.98	97.98	97.75
11	91.01	95.51	96.40	96.85	96.85	96.85	96.63	96.63	96.40	95.96	96.40	96.40	96.85	96.85
12	91.24	95.96	96.85	97.30	97.53	96.85	96.85	96.63	97.30	97.08	96.63	97.75	97.75	97.08
13	91.46	96.85	97.30	97.30	**98.65**	98.20	97.75	97.30	97.08	96.85	96.85	96.85	96.40	96.18
14	91.91	95.73	97.08	96.18	97.08	97.75	97.30	96.63	96.63	96.63	97.08	96.85	96.85	97.08
15	91.91	96.40	96.63	96.85	96.40	96.85	97.30	96.40	96.40	96.63	97.08	97.08	96.85	97.08

Table A.1: Recognition accuracies for particular HMM configurations, best accuracy 98.65% for 13 HMM states and 5 GMM mixture components. Experiment with LBP + Δ^2 features and UWB-06-SLR-A corpus.

The table A.2 shows extended results for recognition of isolated signs. The accuracies of each experiment are presented together with HMM configuration that performed best. The configuration consists of number of HMM states and number of GMM mixture components.

recognition accuracy [%], HMM states, GMM components				feature descriptor
signer dependent		signer independent		
SLR-A	SLR-P	SLR-A	SLR-P	with hand tracking
95.73, 11, 13	84.47, 9, 7	89.69, 9, 10	29.46, 7, 4	HOG hand features
93.03, 11, 13	83.51, 11, 10	92.12, 11, 2	26.87, 11, 8	(x,y) hand coordinates + Δ^2
91.69, 13, 13	-	82.40, 13, 2	-	(x,y) hand c., no hands separation + Δ
91.01, 9, 13	76.44, 7, 8	86.71, 11, 10	20.16, 5, 7	hRDF hand features + Δ
86.07, 13, 13	57.16, 13, 2	90.73, 11, 13	18.92, 11, 4	high level linguistic features
74.61, 11, 13	55.15, 10, 8	75.19, 13, 8	13.93, 9, 10	Hu moments
				without hand tracking
98.65, 13, 5	**93.19**, 6, 8	94.23, 13, 8	27.85, 11, 3	LBP + Δ^2
86.18, 10, 13	72.69, 13, 13	76.08, 9, 10	14.36, 7, 7	DCT
85.53, 13, 13	71.12, 13, 13	75.16, 11, 10	13.95, 8, 9	radon transformation
-	47.78, 8, 2	-	-	eigensigns
74.05, 11, 13	39.59, 12, 13	70.83, 9, 6	7.10, 6, 7	pixel values as features
46.87, 9, 13	43.24, 10, 13	12.27, 8, 9	1.43, 8, 10	AAM
29.37, 9, 13	24.47, 8, 12	8.32, 9, 2	1.34, 7, 5	AAM-ext
				fusion of multiple features
95.51, 12, 10	91.54, 4, 12	**95.06**, 12, 10	32.47, 8, 11	(x,y) hand coordinates + LBP
95.06, 12, 13	84.64, 5, 11	95.05, 13, 11	**34.47**, 5, 3	high level linguistic feat. + LBP
95.73, 8, 8	84.21, 4, 9	90.80, 7, 8	29.46, 4, 4	(x,y) hand coordinates + HOG
95.28, 11, 11	80.63, 3, 11	92.07, 10, 3	29.82, 7, 3	high level linguistic feat. + HOG

Table A.2: Accuracies for recognition of isolated signs, together with number of HMM states and number of GMM mixture components that performed best in every experiment.

The detailed configurations of methods that employ hand tracking are described here:

HOG hand features were computed on hand images normalized to 64px, using 4 orientations and cell size 16x16. Other tested parameters were 8 orientations and cell size 8x8

(x,y) hand coordinates + Δ^2 features have no parameters.

(x,y) hand coordinates, no head/hands separation + Δ this hand tracking is using a different approach for hand tracking and was described in [HKCM11], this tracking is not able to resolve occlusions.

hRDF hand features + Δ was using 72 angle bins and hand images normalized to 64px.

High level linguistic features have no parameters.

Hu moments have no parameters.

Configurations for methods that employ no hand tracking follows:

LBP + Δ^2 the uniform LBP features were directly calculated from the whole image, from which the face region was separated and ignored, without employing the tracking and without using LBP for each hand separately. The LBP radius was set to 3, using 8-neighborhood. Other tested radii were 1,2 and 4.

DCT was using 5x5 coefficient submatrix, other tested values were ranging from 3x3 to 20x20. Additionally, the whole image was split into a 9, 16, 25 and 36 subimages that were processed by DCT separately.

Radon transformation used 20 bins, other tested value was 100 bins. The number of angles used for the radon projection was fixed to 4.

Eigensigns was using 60px normalized images. Other tested values were 40, 80 and 100px. The PCA transformation matrix was trained from 233590 images.

Pixel values as features uses pixel values from a normalized image directly as the features. The resolution of the images was 20px, other tested values were 15 and 10px.

AAM is a set of Active Appearance Model parameters that are used as a feature vector. Thus, this result shows recognition results based on non-manual component only.

AAM-ext has no parameters.

Last summary presents configurations for methods that combine multiple features:

(x,y) hand coordinates + LBP used hand coordinates combined with uniform LBP applied to the hand images normalized to 64px, with radius 2.

High level linguistic features + LBP used the same LBP as above, but combined with high level linguistic features.

(x,y) hand coordinates + HOG combines hand coordinates with HOG features computed for each hand on 64px normalized image, using 4 orientations and 16x16 cells.

High level linguistic features + HOG was using the same HOG features and above but combined with high level linguistic features.