Diplomarbeit im Fach Informatik

An Integrated Tracking And Recognition Approach For Video

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

vorgelegt von:
Jens Forster
Matrikelnummer 245187
E-Mail jens.forster@rwth-aachen.de

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. Dr. rer. nat. T. Seidl

Betreuer:
Dipl.-Inform. Philippe Dreuw
Dipl.-Inform. Thomas Deselaers

20. Mai 2008

Erklärung

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

Aachen, im Mai 2008

Jens Forster

Abstract

This diploma thesis investigates integrated tracking and recognition in continuous sign language recogntion as well as efficient approximations to it. Current state-of-the art sign language recognition system use tracking only as a preprocessing step. Hence, tracking errors lead to recognition errors. We propose to integrate scoring functions of a model-free dynamic programming tracking framework into the visual models of a continuous sign language recognition system. The proposed integrated system as well as approximations to it are evaluated on a publicly available benchmark database for American Sign Language.

Acknowledgment

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

Prof. Dr.-Ing. Hermann Ney for the interesting possibilities at the Chair of Computer Science 6 of the RWTH University Aachen, for the introduction into speech and pattern recognition and for the possibility to attend a workshop on Computer Vision in Bonn;

Prof. Dr. rer. nat. Thomas Seidl who kindly accepted to co-supervise this work,

Philippe "Drö" Dreuw and Thomas "t" Deselaers for their supervision of this work, for many a helpful discussion, for the introduction to image recognition and for their countless ideas, suggestions, effort and assistances this work received;

David Rybach, Björn Hoffmeister and Georg Heigold for explanations regarding the RWTH speech recognizer and lattices;

My former diplomathesis colleagues Volker "oho" Gnann, Peter "Monitor" Fritz, Markus "gnuplot" Nußbaum, Christoph Schmidt, Tobias "tobi" Gass, Tobias "tob" Weyand and Juri "D'Artagnan" Ganitkevitch for many helpful tips, comments and discussions about speech recognition, the RWTH large vocabulary speech recognition system, LaTeX, programming, politics and a lot of relaxing fun;

Patrick Schelenz and Helen Werner for being there when I needed them;

Andreas "husky" Ganser, Patrick Schelenz, Christoph Schmidt, Juri Ganitkevitch, and Markus Nußbaum for partial proof reading.

Of course special thanks go to my parents without whom this work would not have been possible and for supporting my in every way possible. Last but not least, I would like to thank my ex-girlfriend Tanja Kreutz for teaching me a lot about myself during five years and a half, her support, her patience and love as long as it lasted.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object tracking in complex scenes is a challenging problem in many computer vision applications. This is especially true for hand tracking in sign language recognition because the hands sign frequently in front of the speaker's face, interchange, overlap, and may temporally disappear from the scene. Sign language recognition systems commonly apply a two-step procedure to solve the recognition problem. First, features are extracted from the given data and then recognition is performed using the beforehand extracted features.

There are two core problems in this two-step approach to sign language recognition. First, tracking is performed by solving an optimization problem w.r.t. a tracking criterion such as assuming that the hand is the object moving most in an image sequence. Hence, the tracking decision and the extracted features are optimized only according to the chosen tracking criterion and not w.r.t. decisions reached in the recognition phase. Second, possible tracking errors might be impossible to correct in the recognition phase leading to recognition errors because tracking is only performed during feature extraction.

To overcome these problems, we propose a joint tracking and recognition approach. Our goal is to integrate feature extraction and recognition into a single phase with the joint tracking and recognition decision withheld until the end of a given video. Due to the numerous possible positions/locations of an object in an image, the computationally complexity of such an integrated tracking and recognition framework is very high. We investigate several approximations by tracking adaptation and rescoring techniques for the two-step approach easing the computational burden of the integrated tracking and recognition approach.

The overall scientific goal in sign language recognition is to create systems recognizing and translating sign languages to spoken languages and vice versa. Hence, this thesis is concerned with continuous sign language recognition. We test and evaluate the proposed methods in the context of sign language recognition. Nevertheless, methods presented in this work are not limited to sign language recognition.

Isolated sign language recognition is concerned with the recognition of isolated signs without temporal or grammatical context. Typically isolated signs arise from finger

spelling or command and control applications. Recognition of continuous sign language is a challenging task because the appearance of signs partly depends on their temporal context. An overview on the current state of the art in isolated and continuous sign language recognition is presented in Section 1.2.

Furthermore, the lack of available data is a common problem in sign language recognition research leading to recognition systems which generalize poorly. We address this problem using additional virtual training samples leading in training. Moreover, recordings of distinct speakers in sign language databases often vary in scale, illumination, and general positioning of the speaker towards the video camera. Even for a high number of speakers, recognition systems trained on such data often are still speaker dependent. In large vocabulary speech recognition, the problem of speaker dependence is solved by the calculation of time alignments for different speakers in order to minimize effects from gender or speaking rate. We propose to use a visual speaker alignment to gain a speaker independent continuous sign language recognition system.

This thesis is structured as follows: Chapter 2 gives an overview on the continuous sign language recognition system used in the Human Language Technology And Pattern Recognition group at RWTH Aachen University (Cf. [Dreuw & Rybach$^+$ 07]). Visual speaker alignment and the generation virtual training samples in sign language recognition are presented in Chapter 3. An integrated tracking and recognition framework for continuous sign language recognition is developed in Chapter 4 and efficient approximations to the joint tracking and recognition framework are discussed in Chapter 5. Experimental results for joint tracking and recognition, rescoring, and model-driven tracking adaptation using appearance-based features are shown and discussed in Chapter 6. Finally, this work is concluded in Chapter 7.

1.1 State of the art in Tracking

Tracking methods have been applied to many different tasks including gesture recognition, sign language recognition, pedestrian tracking, aerial surveillance, traffic supervision, face recognition, and human movement tracking in general. Among the various proposed tracking methods, three main research areas emerged. These areas are tracking, model-building and tracking by detection, and the combination of tracking and model-driven detection. [Gavrila & Philomin 99] provide a good overview of tracking techniques used in gesture and human movement recognition.

Although a full review of the current state of the art in tracking is far beyond the scope of this work, some of the widely used techniques and applications are described briefly in the following.

Tracking

The Meanshift algorithm, presented in [Comaniciu & Ramesh$^+$ 00], tracks non-rigid objects based on color and texture features. The object to be tracked is characterized using statistical distributions. The algorithm copes with partial occlusions of the object to be tracked, background clutters, rotations in depth and changing camera positions. The continuously adaptive Meanshift (Camshift) algorithm is an extension of Meanshift. In addition to the benefits of Meanshift, Camshift is able to deal with dynamically changing color distributions. [Bradski 98] use Camshift to track human faces in real-time.

In [Isard & Blake 98] the well-known Condensation (conditional density propagation) algorithm is presented. Condensation follows the sequential Monte Carlo approach and estimates the required posterior probability density function by a set of random samples with associated weights. The sequential Monte Carlo approach is also known as particle filtering. An overview of particle filtering with a focus on Kalman filtering and grid based methods is given in [Arulampalam & Maskell$^+$ 02].

Model-based tracking adaptation has been addressed recently by [Wang & Suter$^+$ 07] in the context of particle filtering. They learn a spatial-temporal mixture of Gaussian (SMOG) appearance model and extract histogram features containing spatial and temporal information. The proposed histogram distance between features is used for tracking adaptation. Thus tracking is driven by both the region appearance of target candidates and the region structure.

[Elgammal 05] learn a generative model for a non-linear mapping from a given geometric transformation manifold to the visual input manifold. Tracking is performed by approximating the inverse mapping for each image using Singular Value Decomposition (SVD) [Berrar & Dubitzky$^+$ 03, Chapter 5]. The displacement of the object to be tracked in consecutive images is inferred automatically by this approach. The non-linear manifold mapping approach is reported to be robust to partial occlusion and camera changes.

[Dreuw & Deselaers$^+$ 06] propose a general purpose dynamic programming tracking approach. The actual tracking decision is postponed until all images of a given sequence are processed. The tracking decision is reached by tracing back the best path according to a beforehand specified tracking criterion. The dynamic programming approach is reported to be robust against noise and has been applied successfully in the context of sign language recognition [Dreuw & Rybach$^+$ 07], and automatic video cropping [Deselaers & Dreuw$^+$ 08].

Tracking by Detection

[Ramanan & Forsyth$^+$ 07] propose a two-stage approach that first builds a model of the appearance of individual people and then tracks them by detecting those models

in each frame. For model building, they propose two distinct approaches. First, a bottom-up approach that learns templates for a temporal pictorial structure by clustering using simple edge features is described, and second, a top-down approach detecting stylized poses and learning the pictorial structure accordingly is presented.

An ensemble based tracking by detection scheme is presented in [Avidan 07]. Tracking is considered as a two class classification problem, classifying the object to be tracked against the image background. In each frame a weak classifier is learned choosing object pixels as positive and background pixels as negative examples. Weak classifiers are added to an ensemble over time and a strong classifier is trained using AdaBoost for the ensemble. Tracking between consecutive frames is performed by first classifying every pixel in the new frame using the ensemble classifier resulting in a confidence map of the image. Meanshift is then used to gain a new region of interest from the confidence map and for this region of interest a new weak classifier is trained. A major weakness of this approach is that the object position must be known a-priori for the first frame.

[Okuma & Taleghani$^+$ 04] propose a boosted particle filter for multi-object tracking. Using a cascaded AdaBoost scheme, models for each object are learned on-line. The proposal distribution of the particle filter is formed as a mixture of the AdaBoost detection results and the dynamics model. The system is reported to reliably track multiple objects handling entering and leaving objects.

Integrated Tracking and Detection

[Schiele 06] report a model-free tracking scheme for car surveillance. Homogeneous regions, based on color, position and motion, are extracted and clustered using k-means clustering [Duda & Hart$^+$ 01]. The dominant cluster is then tracked over time and the tracking hypotheses are verified using a Viterbi recognition algorithm.

[Wu & Nevatia 07] propose a multiple person tracking system using a body part model composed of individual part detectors learned using boosting on edgelet features. The part detectors are combined to form a joined likelihood model of all tracked persons. Tracking is performed using data association of the detections of individual part detectors and the detections of the joint likelihood model. If data association can not be performed, Meanshift is used to predict object locations. The system is reported to be robust against partial and full occlusion.

[Andriluka & Roth$^+$ 08] propose a combination of tracking and detection for multiple pedestrian detection and tracking in crowded scenes. They use a dynamical limb model and learn a hierarchical Gaussian process latent variable model based on tracklets. The system is able to reliable track several pedestrians even if longtime occlusion occurs.

1.2 State of the art in Sign Language Recognition

All systems described in this section follow the two-step approach, performing tracking during a feature extraction process. Hence, these systems suffer from the afore mentioned problem of errors in feature extraction.

Research in automatic sign language recognition is performed for both isolated and continuous sign language recognition. Recently, methods from object recognition, e.g. body part models [Ronfard & Schmid$^+$ 02, Felzenszwalb & Huttenlocher 05], [Mikolajczyk & Schmid$^+$ 04], and pose estimation [Sigal & Isard$^+$ 03, Ren & Berg$^+$ 05, Navaratnam & Thayananthan$^+$ 05] are used to enhance sign language systems.

In general, it is difficult to compare available sign language and gesture recognition systems. A main difference are the type of data acquisition, direct-measurement vs. vision-based, and the used evaluation datasets. Direct-measurement is typically performed using data gloves [Fang & Gao 02] or motion capturing systems
[Vogler & Metaxas 01], which provide 3D spatial information of hands, fingers and other body parts with high accuracy and high sampling rates. However, motion capturing and data gloves require complex calibration and preparation, and impose restrictions on the signer's movement. For real world applications data-gloves and motion capturing are not suitable.

Vision-based systems use video cameras for data acquisition and capture the whole signing space. In order to detect the hands and fingers, some systems require the signer to wear colored gloves. Many systems rely on constant illumination, uniform background, and fixed camera and signer positions. Like direct-measurement methods, many of these constraints conflict with natural signing.

In sign language recognition, further problems arise from a lack of common data and from a lack-of-data problem in general. Almost all databases used by the different research groups are not publicly available and differ in language, vocabulary, and size [Ong & Ranganath 05], making a direct comparison of performance almost impossible.

Additionally, one must distinguish between native signers and signers who learned the gestures only for the recording of a database because both groups differ in the manner of signing. Databases recorded by native signers should be used to allow statements on performance in real world applications.

[Ong & Ranganath 05] present an overview of research in sign language, feature extraction and classification methods. Additionally, the authors also analyze inflection, non-manual components, and the grammatical process in sign language.

This work is focused on vision-based sign language recognition. An overview of vision-based systems for isolated and continuous sign language recognition is presented in the remainder of this section.

Isolated Sign Language Recognition

[Assan & Grobel 97] presented one of the first systems for isolated sign language recognition. They use colored gloves for detecting both hands and the fingers of the dominant hand. The used database consists of 262 signs of the sign language of the Netherlands performed by two non-native speakers wearing clothes in background color. A rule-based classifier detects the shoulders and the vertical body axis. The used feature vector includes hand shape, orientation of the hands and fingers, and the hand position normalized w.r.t. shoulder and vertical body axis. The used hidden Markov model classifier achieves a word error rate of 7% to 9% in the speaker dependent case and 8% in the speaker independent case.

[Bowden & Windridge$^+$ 04] use a binary feature vector, called linguistic feature vector, for a two stage classification approach. In the first stage position, shape, and movement are classified and encoded into a 34 dimensional, binary feature vector. In the second stage, independent component analysis (ICA) is applied to project the feature vector into an Euclidean space of lower dimension. The classification itself is performed using Markov chains trained with only one sample of each of the 49 signs (British sign language). The authors report an error rate of 16%.

Dynamic space-time warping using dynamic programming for vision-based character recognition is proposed by [Alon & Athitsos$^+$ 05]. Their database consists of ten digits in Palm's Grafitti Alphabet signed by three different person in three different setups, i.e. with colored gloves, wearing long and short sleeves. The 90 digits recorded with colored gloves are only used for training and model building. The feature vector is composed of color and motion features. The system applies skin detection to find the signers face and then uses motion detection relative to the signers face. The authors report recognition error rates of 9%.

[Cooper & Bowden 07b] stack video frames together as a single volume and search for a trajectory or subspace in the volume corresponding to a sign or gesture. By learning the shape and position of such a subspace they construct a motion detector for isolated sign recognition. The subspace is approximated by a linear combination of simple block features learned through boosting. The system is trained on five repetitions of five different signs performed by nine signers. The test set consists of the same five signs repeated five times by twelve different signers, three of them not being present in the training set. The authors report 90% true positives at 1% false positives.

In [Cooper & Bowden 07a], the authors propose a two-phase viseme based system for "large" lexicon sign language recognition. In the first phase, distinct classifiers are learned for the viseme types placement, movement, and arrangement through boosting. In the second phase, the detection results of the boosted viseme classifiers are combined to a binary feature vector and fed into a first-order Markov chain for each word in the lexicon. The database consists for 164 words (British sign language)

with ten examples each. Trained with five examples per word, the system achieves an error rate of 25%. The database has also been used in [Kadir & Bowden$^+$ 04].

[Zahedi 07] presents an isolated sign language recognition system inferred from a large vocabulary speech recognition system using a dynamic programming tracking framework in preprocessing. Results on various publicly available databases for American, British, Swedish sign language and the sign language of the Netherlands are presented.

Further references for isolated sign language recognition can be found in [Rybach 06] and [Zahedi 07].

Continuous Sign Language Recognition

[Starner & Pentland 94] described one of the first continuous sign language recognition systems. Features for hand position, hand shape, and hand angle are extracted based on a hand segmentation using colored gloves. The system uses hidden Markov models to model words and is trained with 395 sentences of American sign language. The test set consists of 40 sentences with a fixed grammar (pronoun, verb, noun, adjective, same pronoun) and five words each. The authors report a word error rate of 1% for speaker dependent recognition of 99 sentences with grammar restriction and 9% without grammar restriction.

The system of [Starner & Pentland 94] has been extended in [Starner & Weaver$^+$ 98] to use skin color segmentation instead of colored gloves. Two experiments have been carried out using different camera setups. First, a camera showing the signer in frontal view before a complex background and second a head-mounted camera pointing downwards to the hands. Absolute position, movement, and shape features are used for classification. The system is evaluated using 384 sentences for training and 94 sentences for testing. The same grammar restrictions as in [Starner & Pentland 94] apply. The authors report a word error rate of 8% for the frontal view setup and 2% for the head-mounted camera setup being recorded with a different signer.

[Holden & Lee$^+$ 05] describe a tracking algorithm that uses skin color detection and a correspondence algorithm finding the signer's hands and face. The hidden Markov model classifier uses position, shape and hand movements as feature set and achieves a word error rate of 1%. A dataset of 14 distinct sentences of 21 signs in Australian sign language with a fixed grammar is used. The fixed grammar is also used by the recognition system which is trained with 216 out of the 379 utterances. Each of the training sentences occurs in the test set too.

In [Dreuw & Rybach$^+$ 07] a previous version of the system described in this work is presented. Dynamic programming is used in the feature extraction process to track the head and dominant hand of the speaker. The hypothesized positions are used for appearance and vision based feature extraction. 17% word error rate on the RWTH-BOSTON-104 (Cf. Section 6.1) using trajectory and motion features is reported.

Chapter 2

Sign Language Recognition

This work is concerned with vision-based continuous sign language recognition. In contrast to isolated sign language recognition, the number of words in a given video sequence, i.e. sentence, is not known a-priori. Further additional challenges in continuous sign language recognition in contrast to isolated sign language recognition are start-stop detection, *movement epenthesis*, and *coarticulation effects*. Because there are no evident sign boundaries, the start and end of a sign has to be estimated by the recognition system. The reference system of [Dreuw & Rybach$^+$ 07] presented in this chapter and the joint tracking and recognition system presented in Chapter 4 implicitly calculate these boundaries to recognize individual signs. Segments between signs that belong to neither sign are called *movement epenthesis* [Liddell & Johnson 89] and need to be classified as silence or as one of the enclosing signs if no specific model as in [Farhadi & Forsyth$^+$ 07] is trained. Furthermore, the appearance of a sign can change depending on the preceding or succeeding sign. These effects are called *coarticulation effects* and are similar to those observed in speech recognition [Hwang & Hon$^+$ 89]. A survey on current sign language recognition systems can be found in [Rybach 06] and [Ong & Ranganath 05].

This chapter is organized as follows. In Section 2.1 we give an overview on the recognition system used at the Human Language Technology and Pattern Recognition group at RWTH Aachen University. The tracking framework used is described in Section 2.2. The visual model is presented in Section 2.3. Training and Recognition are covered in Sections 2.4 and 2.5.

2.1 System Overview

Most of the state of the art speech recognition and sign language recognition systems use a statistical approach [Jelinek 98]. Given a sequence of input images $X_1^T = X_1, \ldots, X_T$, the best word[1] sequence is chosen according to Bayes' decision rule that

[1] to simplify notation we refer to a sign as a word through out this work

Table 2.1. Symbol reference

Symbol	Description
l	position $l = (x, y)$ at image coordinates (x, y)
X	image represented by U image features $(X^{(1)}, X^{(2)}, \ldots, X^{(U)})$
	$X[l]$ image color/gray value at location l
$\mathcal{R}$	set of offset locations $R = \{(x, y) \mid -\omega \leq x \leq \omega \ \wedge \ -h \leq y \leq h\}$
	for in width range $-\omega, \ldots, \omega$ and in height range $-h, \ldots, h$
Q_t	set of image location $Q_t = \{l_t + l \mid l \in \mathcal{R}\}$
X_R	image sub-part in region Q
X_B	image background $X_B = X \backslash X_R$
$X^{(j)}$	j-th feature extracted from image X
X_1^T	sequence of T images
w_1^N	sequence of N words
w_n	n-th reference word
s_1^T	sequence of T Hidden Markov Model states
l_1^T	sequence of T locations
W	total number of reference words w
T	total time
$S(w)$	total number of Hidden Markov Model states modeling word w

maximizes the *a-posteriori* probability [Bayes 63]:

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{ p\left(w_1^N \mid X_1^T\right) \right\} \tag{2.1}$$

$$= \underset{w_1^N}{\text{argmax}} \left\{ p\left(X_1^T \mid w_1^N\right) \cdot p\left(w_1^N\right) \right\} \tag{2.2}$$

Two basic stochastic models are introduced in Equation (2.2). First, the *visual model* $p\left(X_1^T \mid w_1^N\right)$ models the probability to observe the image sequence X_1^T given the word sequence w_1^N. Second, the *language model* $p\left(w_1^N\right)$ models the *a-priori* probability for a word sequence w_1^N.

The approach is further refined by replacing images X_t by a set of features $x_t := (X_t^{(1)}, \ldots, X_t^{(u)}, \ldots, X_t^{(U)})$ describing the image. Features are extracted at positions l supplied by a tracking framework (Cf. Section 2.2). An overview of the system layout is depicted in Figure 2.1. A survey on and experimental results using the presented system can be found in [Dreuw & Rybach$^+$ 07]. For simplicity, the notation X_t is used throughout this work unless the more detailed formulation is needed.

The focus of this work is an integrated tracking and recognition approach. New approaches to an integrated tracking and recognition framework for continuous sign language recognition are described in Chapters 4 and 5.

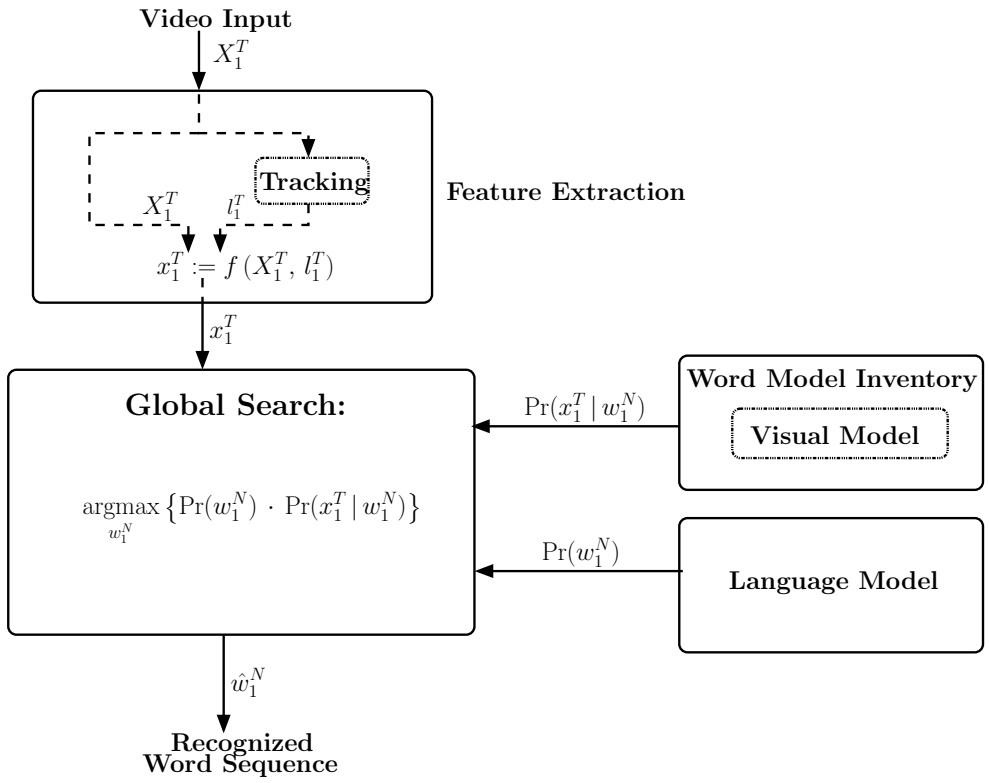


Figure 2.1. Basic structure of a continuous sign language system based on a statistical speech recognition system

2.2 Tracking

In the described sign language recognition system, object tracking is performed during feature extraction and is considered as a separate optimization problem. The sequence of object positions l_1^T is searched that maximizes the likelihood of this sequence given the image sequence X_1^T:

$$[l_1^T]_{\text{opt}} = \underset{l_1^T}{\operatorname{argmax}} \{p\left(l_1^T \mid X_1^T\right)\} \tag{2.3}$$

$$= \underset{l_1^T}{\operatorname{argmax}} \left\{ \prod_{t=1}^{T} p\left(l_t \mid l_1^{t-1}, X_1^t\right) \right\} \tag{2.4}$$

Assuming a first-order Markov process, Equation (2.4) can be further simplified:

$$[l_1^T]_{\text{opt}} = \underset{l_1^T}{\operatorname{argmax}} \left\{ \prod_{t=1}^{T} p\left(l_t \mid l_{t-1}, X_{t-1}^t\right) \right\} \tag{2.5}$$

$\log(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right))$ can be expressed by an image dependent scoring function $q\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$ that rates an object position l_t with a score depending on the previous positions l_{t-1} and the images X_{t-1}^t, and an image independent scoring function $\mathcal{J}(l_t,\, l_{t-1}) = \parallel l_t - l_{t-1} \parallel^2$ controlling properties of the tracked object position sequence.

$\mathcal{J}(l,\, l')$ is called *jump penalty*. It penalizes large distances between consecutive object positions because it is not likely that succeeding object positions are far away from each other. Using the score functions $q\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$ and $\mathcal{J}(l_t,\, l_{t-1})$, the optimization criterion (2.5) can be rewritten to:

$$[l_1^T]_{\text{opt}} = \underset{l_1^T}{\operatorname{argmax}} \left\{ \sum_{t=1}^{T} q\left(l_t,\, l_{t-1},\, X_{t-1}^t\right) - \mathcal{J}(l_t,\, l_{t-1}) \right\} \tag{2.6}$$

The jump penalty is subtracted from the score function in order to penalize wide movements. The described system uses the squared Euclidean distance as jump penalty function.
The optimization problem (2.6) is solved using dynamic programming leading to the following recursion equations

$$S(t,\, l) = \max_{l' \in \mathcal{M}(l)} \left\{ S(t-1,\, l') - \mathcal{J}(l,\, l') + q(l,\, l',\, X_{t-1}^t) \right\} \tag{2.7}$$

$$B(t,\, l) = \underset{l' \in \mathcal{M}(l)}{\operatorname{argmax}} \left\{ S(t-1,\, l') - \mathcal{J}(l,\, l') + q(l,\, l',\, X_{t-1}^t) \right\} \tag{2.8}$$

where $S(t,\, l)$ stores the score for the best path ending in position l at time t, $\mathcal{M}(l)$ is the set of possible predecessor positions l' to position l and $B(t,\, l)$ stores the predecessor of position l at time $t-1$.
The best position sequence is hence traced back as follows:

1. Search best ending position:

$$l_T = \underset{l}{\operatorname{argmax}} \left\{ S(T,\, l) \right\} \tag{2.9}$$

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

$$l_t = B(t+1,\, l_{t+1}) \tag{2.10}$$

The described tracking framework is a multi-purpose tracking framework. Depending on the used scoring function, different objects can be tracked. Therefore, the framework is not limited to sign language recognition. In [Deselaers & Dreuw$^+$ 08] the framework has been used for time-coherent, trained automatic video cropping.

In this work the tracking framework is used for tracking the dominant hand. The main assumption is that the hand is the only or strongest moving object in a video sequence. Hence, scoring functions (Cf. Figure 4.4) modelling motion

$$q(l_t,\, l_{t-1},\, X_{t-1}^t) = \sum_{l \in Q_t} \left(X_t'[l]\right)^2 \tag{2.11}$$

or a constant background and motion history

$$\begin{aligned} -\, q(l_t,\, l_{t-1},\, X_{t-1}^t) \;&=\; \alpha_1 \sum_{l \in Q} \left(X_t[l_{t-1}+l] + X_{t-1}[l_{t-1}+l]\right)^2 \\ &+\; \alpha_2 \sum_{l \notin Q_t \cup Q_{t-1}} \left(X_t'[l]\right)^2 \end{aligned} \tag{2.12}$$

are suitable for tracking the dominant hand, with $X_t' = X_t - X_{t-1}$ and weighting factors α_1 and α_2. Tracking of the dominant hand is performed using the constant background and motion history scoring function in this work.

For a full survey on the used tracking framework see [Dreuw & Deselaers$^+$ 06] and [Rybach 06].

2.3 Visual Modeling

Visual models provide stochastic models of signs capturing static as well as temporal features of sign language.

In large vocabulary speech recognition systems acoustic models model words by sub-word units such as phonemes. Sub-word units are shared between all words in the given vocabulary. Whole word models are created by connecting sub-word models according to a phonetic lexicon. By using sub-word models it is possible to recognize words, for which no example is given in the training data of the system, by providing knowledge about the pronunciation of the word. Furthermore, sub-word models can be estimated more reliably because more training data is available for each sub-word.

Unfortunately, no proper and/or broadly accepted definition of sub-word units for sign language is available until now. The usage of self-organizing sub-word units has been investigated for isolated sign language by [Bauer & Kraiss 01, Bauer & Kraiss 02]. The sub-word units are defined using a data-driven approach without linguistic models. [Bauer & Kraiss 02] report an error rate of 7% on 100 signs with 150 sub-word units. 50 unknown signs, automatically transcribed using trained sub-word units, are reported to be classified with an error rate of 19%. The usage of self-organizing sub-word models for continuous sign language recognition is discussed in [Bauer]. The system presented in the following uses whole word models.

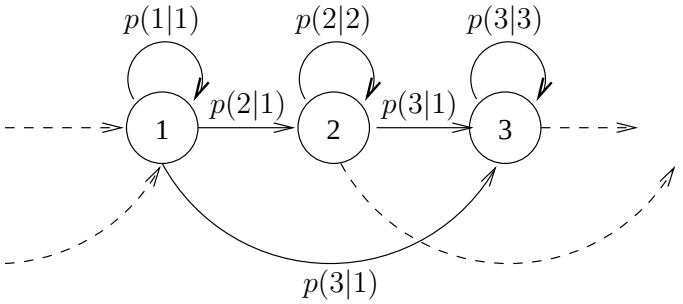


Figure 2.2. Three state hidden Markov model in Bakis topology

Hidden Markov models (**HMMs**) are stochastic finite state automata. Each part of a word is represented by states of the automaton. The states of a HMM are an abstract concept and can not be observed in the given data, i.e. they are *hidden*. The system presented in this work uses the 0-1-2 or Bakis topology [Bakis 76]. Using the Bakis topology, each state has three outgoing transitions: a loop transition (0-transition) to stay in the current state, a forward transition (1-transition) to the next state and a skip transition (2-transition) to the state after the next one. The Bakis topology is depicted in Figure 2.2. If sub-word units are available, HMMs for sub-words are concatenated to obtain HMMs for whole words.

In the context of the HMM approach, the probability $p\left(X_1^T \,|\, w\right)$ of observing an image sequence X_1^T given a word w is defined as the marginal distribution with respect to all possible state sequences s_1^T for this word:

$$p\left(X_1^T \,|\, w\right) = \sum_{[s_1^T]} p\left(X_1^T,\, s_1^T \,|\, w\right) \tag{2.13}$$

$p\left(X_1^T,\, s_1^T \,|\, w\right)$ can be rewritten to:

$$p\left(X_1^T,\, s_1^T \,|\, w\right) = \prod_{t=1}^{T} p\left(X_t,\, s_t \,|\, X_1^{t-1},\, s_1^{t-1},\, w\right) \tag{2.14}$$

Whole word HMMs are concatenated to model a whole word sentence. Using Equations (2.13) and (2.14), the probability of observing an image sequence X_1^T given a sentence w_1^N can be written as:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \prod_{t=1}^{T} p\left(X_t,\, s_t \,|\, X_1^{t-1},\, s_1^{t-1},\, w_1^N\right) \tag{2.15}$$

In Equation (2.15), the marginal distribution is calculated over all possible state sequences s_1^T for a given sentence w_1^N. Using Bayes' identity, Equation (2.15) can further be rewritten to:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \prod_{t=1}^{T} p\left(X_t \,|\, X_1^{t-1},\, s_1^t,\, w_1^N\right) \cdot p\left(s_t \,|\, X_1^{t-1},\, s_1^{t-1},\, w_1^N\right) \tag{2.16}$$

Assuming that observing X_t depends only on the current state s_t, and that state s_t depends only on the direct preceding state s_{t-1} (*first order Markov assumption*), Equation (2.16) can be simplified to:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \prod_{t=1}^{T} p\left(X_t \,|\, s_t,\, w_1^N\right) \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \tag{2.17}$$

Using Equation (2.17), the marginal distribution $p\left(X_1^T \,|\, w_1^N\right)$ is split into an *emission probability* $p\left(X_t \,|\, s_t,\, w_1^N\right)$ and a *state transition probability* $p\left(s_t \,|\, s_{t-1},\, w_1^N\right)$. The emission probability specifies the probability of observing image X_t in state s_t. The probability to move from state s_{t-1} to state s_t is expressed by the state transition probability (Cf. Figure 2.2). The sum in Equation (2.17) is approximated by the maximum. Experimental results show that this *Viterbi* or *maximum approximation* is comparable to calculating the full sum [Ney 90].

$$p\left(X_1^T \,|\, w_1^N\right) \approx \max_{[s_1^T]} \left\{ \prod_{t=1}^{T} p\left(X_t \,|\, s_t,\, w_1^N\right) \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\} \tag{2.18}$$

Using either the *forward-backward algorithm* [Baum 72, Rabiner & Juang 86] or *dynamic programming* [Bellmann 57, Viterbi 67, Ney 84], Equations (2.17) and (2.18) can be evaluated efficiently.

Assuming independence of the state transition probability from the enclosing word model, the state transition probability can be replaced by an auxiliary function $q\left(s_t - s_{t-1}\right)$. This auxiliary function is commonly called *time distortion penalty* (**TDP**). In case of the used Bakis topology the auxiliary function q has to be defined for $q\left(0\right)$ (loop transition), $q\left(1\right)$ (forward transition) and $q\left(2\right)$ (skip transition). In the presented system TDPs are defined via fixed values instead of being estimated on the training data.

$$p\left(X_1^T \,|\, w_1^N\right) \approx \max_{[s_1^T]} \left\{ \prod_{t=1}^{T} p\left(X_t \,|\, s_t,\, w_1^N\right) \cdot q\left(s_t - s_{t-1}\right) \right\} \tag{2.19}$$

2.4 Training

The presented system uses stochastic models as knowledge sources. The true distributions of the language and visual model are not known and must be estimated from training data.

Gaussian mixture densities are used to model the emission probabilities $p(X \mid s, w)$. The model parameters mean μ_{esw} and variance Σ have to be estimated. In order to simplify notation, the model parameters are summarized as a parameter set ϑ. In the following a training set of R pairs is assumed for training. Each training pair consists of an image sequence $[X_1^{T_r}]_r$ and a corresponding transcription $[w_1^{N_r}]_r$.

The training data is processed sentence-wise. Additionally, the word boundaries are unknown and have to be estimated. The estimation of the word boundaries is performed implicitly by constructing a super HMM for each sentence by concatenating the individual word HMMs.

The model parameter set ϑ is estimated using the maximum likelihood criterion via an *expectation maximization* (EM) algorithm. The parameter set $\hat{\vartheta}$ is searched that maximizes the likelihood of the training data:

$$\hat{\vartheta} = \underset{\vartheta}{\operatorname{argmax}} \left\{ \prod_{r=1}^{R} p\left(\left[X_1^{T_r} \right]_r \mid \left[w_1^{N_r} \right]_r, \vartheta \right) \right\} \tag{2.20}$$

Because of the used Viterbi approximation (Cf. Equation (2.18)), an image contributes to exactly one emission probability $p(X \mid s, w)$. The training is performed using the following algorithm:

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

The training algorithm is iterated until either an alignment remains stable or a beforehand fixed number of iterations is reached.

The described training algorithm needs an initial alignment to estimate initial model parameters. *Linear segmentation* is used for the initialization. First, the image sequence X_1^T is is split into three parts:

$$\underbrace{X_1 \dots X_{b-1}}_{\text{silence}} \quad \underbrace{X_b \dots X_e}_{\text{speech}} \quad \underbrace{X_{e+1} \dots X_T}_{\text{silence}}$$

The *start-stop detection* searches optimal word boundaries b and e by minimizing the log-likelihood of the segments using two Gaussian densities. One Gaussian densitiy

models silence and the other models speech [Bridle & Sedgwick 77]. A problem specific to sign language recognition is how to detect silence. Further information on this topic can be found in [Rybach 06]. Images, which are classified to be silence, are assigned to the silence model while images in the detected speech segment are linearly aligned to the given HMM. Using these aligned images, initial model parameters are estimated.

To increase the number of densities, successive splitting of the mixture densities is applied [Ney 05, Chapter 2.2]. By splitting the mixture densities the estimated probability density models the true probability density more smoothly.

In contrast to the training of the visual model, only the transcriptions of the training data are used to train the language model. In the presented system the SRILM toolkit [Stolcke 02] is used to estimate language model parameters with Kneser-Ney smoothing.

2.5 Recognition

The recognition process aims to find a word sequence $[w_1^N]_{\text{opt}}$ which maximizes the posterior probability given an image sequence X_1^T. Given a language model history h_n, the recognition process can be formulated as the following optimization problem:

$$\begin{aligned}
[w_1^N]_{\text{opt}} &= \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(w_1^N \mid X_1^T\right) \right\} \\
&= \underset{w_1^N}{\operatorname{argmax}} \left\{ \left[\prod_{n=1}^{N} p\left(w_n \mid h_n\right) \right] \cdot \max_{s_1^T} \left\{ \prod_{t=1}^{T} p\left(X_t \mid s_t,\, w_1^N\right) \cdot p\left(s_t \mid s_{t-1},\, w_1^N\right) \right\} \right\} \\
&= \underset{w_1^N}{\operatorname{argmax}} \left\{ \left[\prod_{n=1}^{N} p\left(w_n \mid h_n\right) \right] \right. \\
&\quad \left. \cdot \max_{s_1^T} \left\{ \prod_{t=1}^{T} p\left(X_t^{(1)}, \ldots, X_t^{(U)} \mid s_t,\, w_1^N\right) \cdot p\left(s_t \mid s_{t-1},\, w_1^N\right) \right\} \right\} \\
&= \underset{w_1^N}{\operatorname{argmax}} \left\{ \left[\prod_{n=1}^{N} p\left(w_n \mid h_n\right) \right] \right. \\
&\quad \left. \cdot \max_{s_1^T} \left\{ \prod_{t=1}^{T} p\left(f_1(X_t,\, l_t), \ldots, f_U(X_t,\, l_t) \mid s_t,\, w_1^N\right) \cdot p\left(s_t \mid s_{t-1},\, w_1^N\right) \right\} \right\} \qquad (2.21)
\end{aligned}$$

Using *Viterbi search*, the optimization problem can be solved using dynamic programming [Ney 84].

At each time step all HMM states for all current hypothesis are expanded, i.e. the succeeding HMM states are calculated. A hypothesis represents the sequence of words recognized so far up to the current time step. Additionally, the path of HMM states

in the current word is part of the hypothesis. The likelihood of all hypotheses can be directly compared because Viterbi-search is time synchronous. Furthermore unlikely hypotheses can be discarded. The discarding of unlikely hypothesis is called *pruning* [Ney & Mergel$^+$ 87].

Since the described reference system extracts features $X_t^{(1)}, \ldots, X_t^{(U)}$ for each time frame during preprocessing according to an optimization/tracking criterion different from Equation (2.21), errors in feature extraction lead to features fitting the trained object models poorly. The recognition problem of Equation (2.21) uses only the beforehand extracted features. Thus, errors in feature extraction lead to errors in recognition which can not be compensated by the system. To compensate for errors in feature extraction we propose to fuse feature extraction, e.g. tracking of body parts, and recognition into an integrated optimization problem for recognition. Due to the additional knowledge source, errors in feature extraction can be corrected in such an integrated system. Additionally, tracking decisions can be optimized w.r.t. the hypothesized word sequence. An integrated tracking and recognition system for continuous sign language recognition is described in Chapter 4.

Chapter 3

Model Enhancement

In this chapter, we propose two model enhancement techniques to overcome the common lack-of-data problem in continuous sign language recognition.

Continuous sign language recognition systems need to cope with various difficulties to recognize sign language speaker independently. The appearance of a sign depends on it's preceding and succeeding sign. Furthermore, the appearance of a sign can change significantly in different utterances of the same speaker or in utterances of different speakers. Statistical recognition systems need a large amount of training data to model these variabilities, and to estimate all model parameters reliably. Publicly available sign language databases commonly feature only a low number of different speakers, small vocabularies, constrained grammars, a high number of singletons, and a small number of sentences. Therefore, some visual models are estimated on only few observations per density. Even "one-shot" training is necessary for singletons. This results in too sharp Gauss density means which do not generalize well on unseen data. We propose to generate virtual training samples from the existing training data accounting for translation effects and effectively increasing the amount of similar training data.

Moreover, recordings of different speakers often vary in illumination, scale, and general positioning of the speaker towards the camera in publicly available sign language databases. Statistical sign language recognition systems trained using appearance-based features essentially learn the differences between different speakers. Hence, most statistical sign language recognition systems are speaker dependent. Using appearance-based features, a continuous sign language recognition system can be trained to recognize sign language speaker independently if bodies of the signing speakers have the same baseline depth and scale. Therefore, we propose visual speaker alignment to transform the bodies of different speakers to the same scale and baseline depth.

In the following, visual speaker alignment and the generation of virtual training samples are discussed in Sections 3.1 and 3.2 respectively.

3.1 Visual Speaker Alignment

Visual speaker alignment (VSA) adapts the bodies of the signing speakers to the same scale and baseline depth in a video frame. Recordings of different speakers in sign language databases often vary in illumination, scale and general positioning of the speakers to the camera leading to essentially speaker dependent recognition systems. Systems trained using appearance-based features on such training data learn the differences between different speakers and not the characteristic appearance of a sign. Speaker dependent speaker adaptation is used in continuous speech recognition (ASR) to reduce the phonetic variability of spoken words in utterances of the same speaker and in utterances of different speakers [Woodland 01]. ASR systems trained using speaker dependent speaker adaptation recognize speech speaker independently. Similar to speaker dependent feature adaptation in ASR, vision-based feature adaptation or VSA has been proposed by [Dreuw & Ney 08] for continuous sign language recognition. We use the method described by [Dreuw & Ney 08] to visually align speakers in the RWTH-BOSTON-104 database.

In the following, we briefly describe VSA.

[Dreuw & Ney 08] use the Viola & Jones head detection method [Viola & Jones 04] to detect the face of each speaker in a reference recording of each speaker. Figure 3.1 in the top row shows one cropped region-of-interest Q_n of the reference recording of each speaker. The detected face r_n of speaker n is depicted by the turquoise rectangle.

For every speaker n of a set of $1, \ldots, N$ a speaker dependent affine warping matrix A_n is estimated so that the difference between all overlapping cropped regions-of-interest, and their corresponding detected faces is minimal. Given a region-of-interest Q_n (Cf. Figure 3.2) and a face detection rectangle $r_n(x, y, w, h) := \{(x - \frac{w}{2},\, y - \frac{h}{2}),\, (x + \frac{w}{2},\, y + \frac{h}{2})\}$, the affine warping matrices A_m of the remaining $N - 1$ speakers are optimized w.r.t. the warped cropped regions-of-interest Q'_n and Q'_m, and w.r.t. a face penalty function $q(r'_n, r'_m)$ which penalizes large differences in face position and ratio of the warped face detection rectangles. Given binary thresholded images $\tilde{X}_n$ and $\tilde{X}_m$, VSA is expressed by the following optimization criterion:

$$\min_{A_n,\, A_m} \left\{ \sum_{\substack{u \in Q'_n \\ u' \in Q'_m}} \left(\tilde{X}_n[u] - \tilde{X}_m[u'] + \alpha \cdot q(r'_n,\, r'_m) \right) \right\} \tag{3.1}$$

The resulting aligned regions-of-interests are shown in the lower row of Figure 3.1. The dotted turquoise lines depict the corresponding warping transformation. Comparing the images of the first row before VSA with the images of the second row after VSA in Figure 3.1, the scale variance between different speakers is reduced. All speakers are

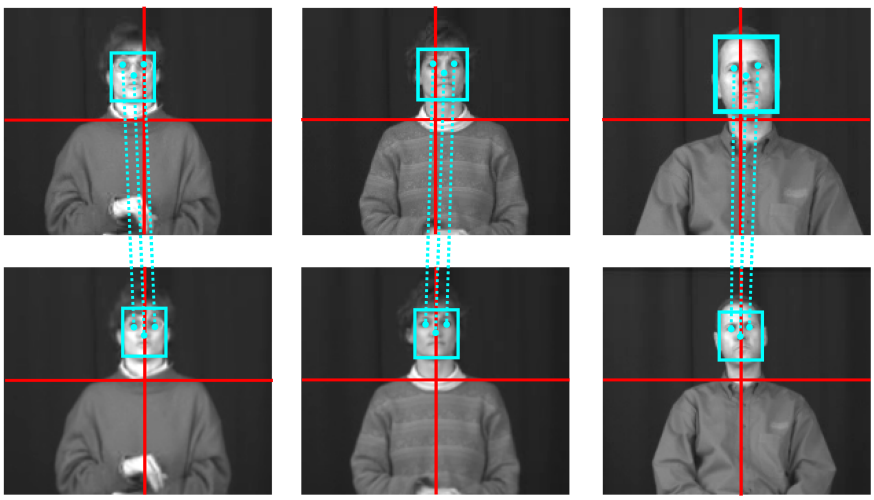


Figure 3.1. Visualization of Automatic Visual Speaker Alignment

centered in the region-of-interest after VSA and all shoulders lie on the same height level depicted by the horizontal red line.

3.2 Virtual Training Samples

We propose to generate virtual training samples (VTS) from the given training data to overcome the common lack-of-data problem in continuous sign language recognition. Statistical speech and sign language recognition system need a high amount of training data to robustly model coarticulation effects, and inter- and intra-personal variability in sign appearance. A lack of available data leads to a low number of observations used to train individual Gaussian densities in the visual models leading to visual models which do not generalize well on unseen data. For other pattern recognition tasks it has been shown that the performance of a statistical recognition system can be significantly improved by using additional training data [Burges & Schöllkopf 97]. In the context of sign language recognition, [Wang & Chen$^+$ 06] recently proposed resampling in Chinese sign language recognition to overcome the lack-of-data problem.

In this work, we generate VTS as described in [Dreuw & Forster$^+$ 08].

Figure 3.2 shows a sample frame from the RWTH-BOSTON-104 database (Cf. Section 6.1). We crop only a region-of-interest (ROI) from each frame in the database because the lower part of each image contains additional textual information about the frame, and the left and right borders are unused. The ROI is depicted by the solid yellow rectangle in Figure 3.2. We shift the ROI center (x, y) by δ pixels in x- and

Figure 3.2. Generation of Virtual Training Samples

y-direction. Thus, we account for small translations in each direction and add variety to the training set. The shifted ROIs are depicted by the transparent yellow rectangles in Figure 3.2. In this work, we shift the ROI by $\delta = \pm 1$ Pixel enlarging the training set by a factor of nine.

The described VTS generation can be interpreted as a distortion and adaptation on the signal level. Each additional VTS may lead to a slightly different tracking path and hence to a different feature sequence in training.

Chapter 4

Integrated Tracking And Recognition

Combined recognition systems have been proposed for various classification problems. [Kobayashi & Haruyama 97] propose partly-hidden Markov models for gesture recognition. In partly-hidden Markov models one observable state alternates with a hidden state. Two dimensional HMMs are used by [Othman & Aboulnasr 00] for face recognition to allow temporal and spatial transitions between HMM states.

Recently, [Chu & Huang 07] proposed coupled HMMs for audio-video speech recognition. In contrast to two dimensional HMMs, they use two separate HMMs for speech and vision tightly coupling these HMMs by windowing. Hidden state shape models (HSSM) are used by [Wang & Athitsos$^+$ 07] to model objects with varying structure such as tree leaves.

[Dreuw & Rybach$^+$ 07] present a first approach to combined tracking and recognition for continuous sign language recognition by explicitly fusing tracking score calculation with the systems emission score. In this work, we further refine this approach.

Here, we propose to integrate the tracking and the recognition process, and to simultaneously solve the tracking and recognition problem using time synchronous linear search. The proposed combined system postpones the tracking and recognition decision until a complete sentence is processed. Thus, the chosen tracking path is optimized according to the hypothesized word sequence and not w.r.t. a beforehand specified tracking criterion. In contrast to the reference baseline system (Cf. Chapter 2), tracking errors can be corrected in the joint optimization problem. The structure of the proposed integrated tracking and recognition system is depicted in Figure 4.1.

This chapter is structured as follows: In Section 4.1 a short overview of linear search as used in automatic speech recognition is given. Visual modeling and the tracking model for combined linear search in continuous sign language recognition are addressed in Sections 4.2 and 4.3 respectively. We derive the optimization problem resulting from combined linear search in Section 4.4 and solve it using dynamic programming in Section 4.5. The resulting time and space complexities are discussed in Section 4.6. The chapter is concluded by an overview of the new combined system in Section 4.8.

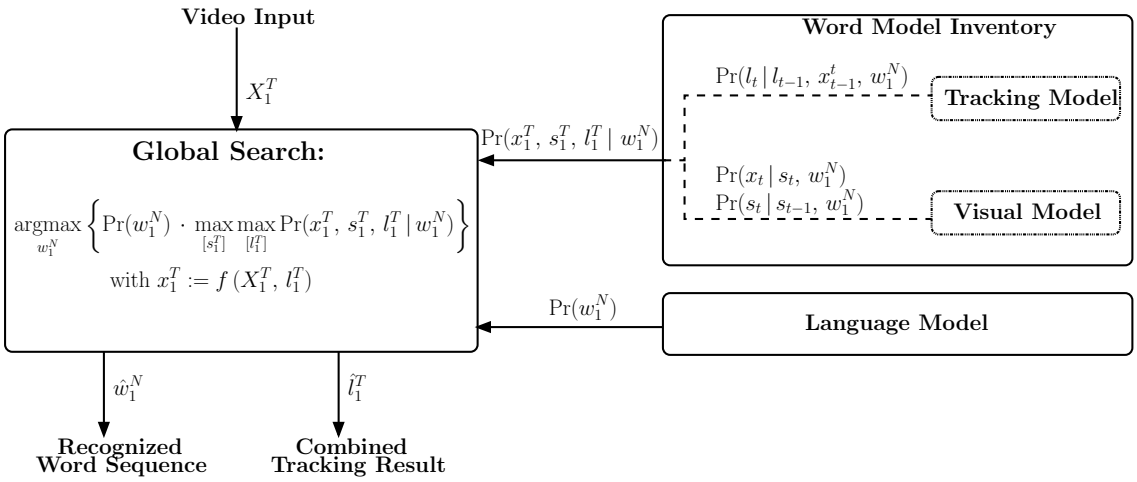


Figure 4.1. Basic structure of an integrated tracking and recognition system

4.1 Overview of Linear Search

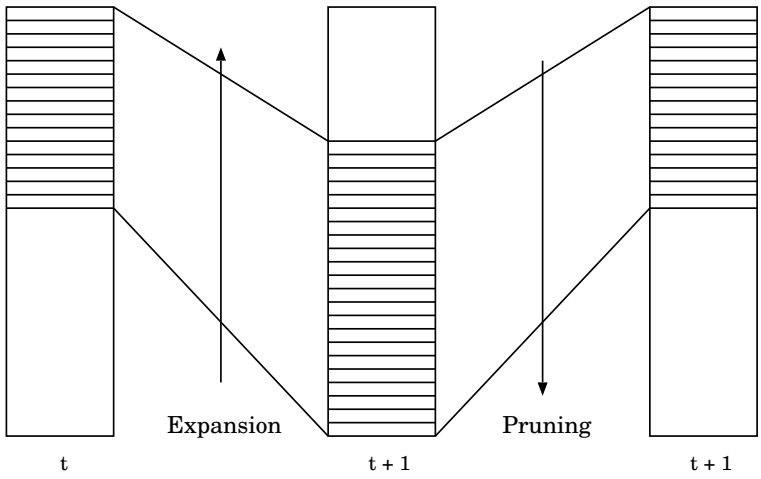
Linear search solves the recognition problem in speech and sign language recognition by unfolding the HMM over time and aligning it to the corresponding language model probability.

In contrast to other search techniques used in speech recognition, e.g. word conditioned tree search, linear search hypothesizes all reference words to be possible ending and starting at each time frame t. Hence, a word-end recombination has to be performed at each time frame for every reference word. The set of state hypotheses is expanded according to the chosen HMM topology from time frame t to $t+1$ and consequently pruned according to a given pruning criterion. Commonly used pruning criteria are value thresholding (pruning a hypothesis if the probability of state is lower than a threshold) and histogram pruning (limiting the overall number of state hypotheses at a given time frame). The process of hypotheses expansion from time frame t to $t+1$ is shown in Figure 4.2a.

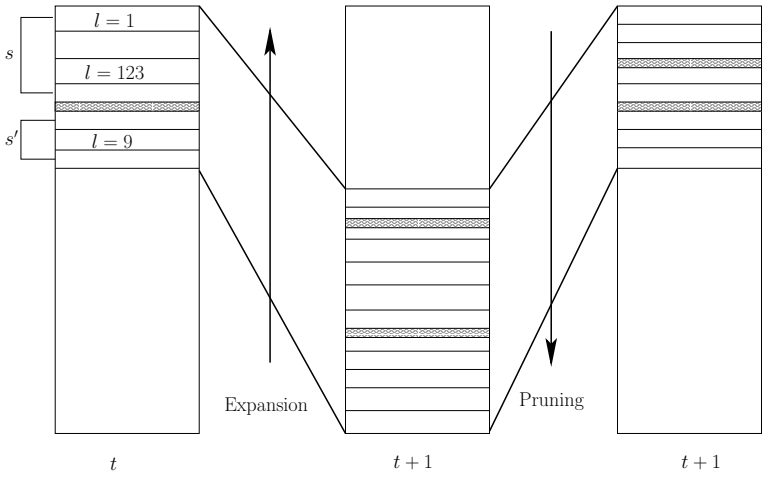
4.2 Visual Modeling

In order to combine tracking and recognition into a joint optimization problem, we need to incorporate spatial information into the recognition process. The main idea is to combine the HMM emission probabilities with the tracking scoring functions. We revise the global decision rule of Equation (2.2)

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(X_1^T \mid w_1^N\right) \cdot p\left(w_1^N\right) \right\} \tag{4.1}$$



(a): Linear Search [Ney 05]; rectangle represents a single HMM state



(b): Combined Linear Search; rectangle represents a combination of state and location i.e. state s at location $l = 1$

Figure 4.2. Hypotheses Expansion from time t to $t + 1$, first all HMM states are expanded according to the chosen topology and then pruning removes unlikly state expansions

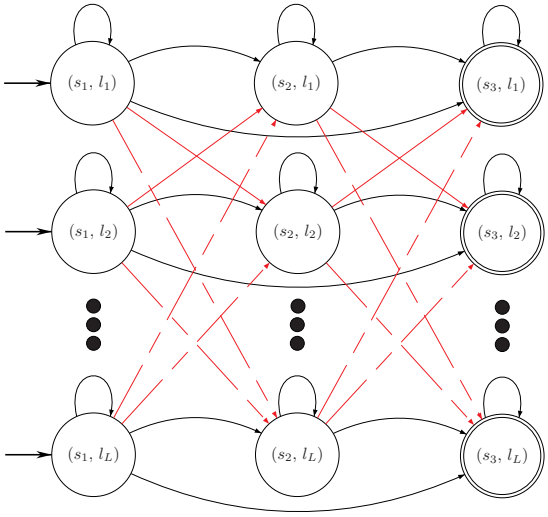


Figure 4.3. Combined HMM

and model spatial information, i.e. image tracking locations, as additional hidden variable l in the system's visual model by taking the marginal distribution over the image locations.

$$p\left(X_1^T \,|\, w_1^N\right) \quad = \quad \sum_{[s_1^T]} \sum_{[l_1^T]} p\left(X_1^T,\, s_1^T,\, l_1^T \,|\, w_1^N\right) \tag{4.2}$$

By explicitly modelling HMM states s_1^T *and* image locations l_1^T as hidden variables in Equation (4.2), the hypotheses structure of the continuous sign language recognition system changes from single HMM states to tuples of states and locations. We use the Bakis topology of transitions between states but apply no restrictions on spatial movement. The new hypotheses structure is shown in Figure 4.3 with additional transition edges depicted in red.

$p\left(X_1^T,\, s_1^T,\, l_1^T \,|\, w_1^N\right)$ can be rewritten to:

$$p\left(X_1^T,\, s_1^T,\, l_1^T \,|\, w_1^N\right) = \prod_{t=1}^{T} p\left(X_t,\, s_t,\, l_t \,|\, s_1^{t-1},\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \tag{4.3}$$

Using Equations (4.2) and (4.3), the probability of observing an image sequence X_1^T given a whole word sentence w_1^N can be written as:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \sum_{[l_1^T]} \prod_{t=1}^{T} p\left(X_t,\, s_t,\, l_t \,|\, s_1^{t-1},\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \tag{4.4}$$

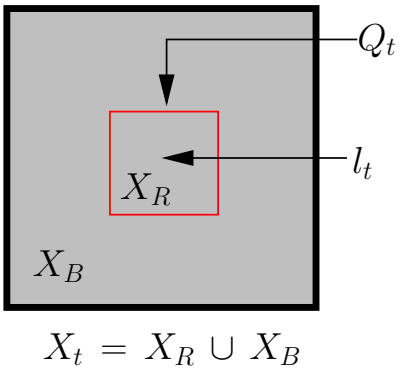


Figure 4.4. Region of interest spanned by an image location l

Note that in Equation (4.4) the marginal distribution is calculated over the state sequence s_1^T and the sequence of image locations l_1^T. Using Bayes' identity, Equation (4.4) is rewritten to:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \sum_{[l_1^T]} \prod_{t=1}^{T} \left(p\left(X_t,\, l_t \,|\, s_1^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \cdot\; p\left(s_t \,|\, s_1^{t-1},\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \right) \tag{4.5}$$

Assuming state s_t depends only on its preceding state s_{t-1} and the word sequence w_1^N, i.e. first order Markov assumption, Equation (4.5) is simplified to:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \sum_{[l_1^T]} \prod_{t=1}^{T} \left(p\left(X_t,\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right) \tag{4.6}$$

$p\left(s_t \,|\, s_{t-1},\, w_1^N\right)$ is denoted as *state transition probability* (Cf. Section 2.3).

In the following an image X_t is represented by a set of U image features $(X_t^{(1)}, \ldots, X_t^{(u)}, \ldots, X_t^{(U)})$ for an object to be tracked. Given a candidate location l_t for the object to be tracked, arbitrary many features can be extracted using suitable feature extraction functions $f_1, \ldots, f_u, \ldots, f_U$ where $X_t^{(u)} = f_u\left(X_t,\, l_t\right)$ is extracted from the region of interest Q_t around l_t. Feature extraction for a given feature extraction function f and a candidate location l_t is shown in Figure 4.4. We abbreviate a subset of features of an image X_t by $X_t^{\left(\,\frac{U}{2}\,\right)} = (X_t^{(2)}, \ldots, X_t^{(U)})$.

Thus, Equation (4.6) is rewritten to:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{[s_1^T]} \sum_{[l_1^T]} \prod_{t=1}^{T} \left(p\left(X_t^{(1)},\, X_t^{(\frac{U}{2})},\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right) \tag{4.7}$$

As depicted in Figure 4.4, the region of interest Q_t segments the image X_t into a foreground image $X_{R,\,t}$ and a background image $X_{B,\,t}$. The background image $X_{B,\,t}$ is implicitly determined by the foreground image $X_{R,\,t}$ which is defined at globally best location l_t. Therefore, we only focus on the foreground image $X_{R,\,t}$ and feature $X_{R,\,t}^{(1)}$.

In contrast to the formulation of the emission probability derived in Section 2.3, $p\left(X_{R,\,t}^{(1)},\, X_t^{(\frac{U}{2})},\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right)$ contains spatial information and needs to be simplified to be evaluated efficiently. Using Bayes' identity twice, this probability is rewritten as:

$$\begin{aligned} p\left(X_{R,\,t}^{(1)},\, X_t^{(\frac{U}{2})},\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) = \\ p\left(X_{R,\,t}^{(1)} \,|\, s_{t-1}^t,\, l_1^t,\, X_t^{(\frac{U}{2})},\, X_1^{t-1},\, w_1^N\right) \cdot p\left(l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_t^{(\frac{U}{2})},\, X_1^{t-1},\, w_1^N\right) \cdot \\ p\left(X_t^{(\frac{U}{2})} \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) \end{aligned} \tag{4.8}$$

Since image features $X_t^{(u)}$ are extracted from image X_t using a feature extraction function $f_u(X_t,\, l_t)$ depending on the candidate object location l_t, the probability $p\left(X_t^{(\frac{U}{2})} \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right)$ is constant for all hypothesized word sequences and hence omitted in the following calculation.

Moreover, we set $U = 2$ and define image feature $X_t^{(2)}$ as the original image X_t, i.e. $f_2\left(X_t,\, l_t\right) = X_t$, to comply with the formulation of the tracking frame work presented in Chapter 2. Equation (4.8) is hence rewritten to

$$\begin{aligned} p\left(X_{R,\,t}^{(1)},\, X_t,\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) = \\ p\left(X_{R,\,t}^{(1)} \,|\, s_{t-1}^t,\, l_1^t,\, X_1^t,\, w_1^N\right) \cdot p\left(l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^t,\, w_1^N\right) \end{aligned} \tag{4.9}$$

Assuming that candidate location l_t depends only on its preceding location l_{t-1} and the images X_{t-1}^t, Equation (4.9) is further simplified to:

$$\begin{aligned} p\left(X_{R,\,t}^{(1)},\, X_t,\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) = \\ p\left(X_{R,\,t}^{(1)} \,|\, s_{t-1}^t,\, l_1^t,\, X_1^t,\, w_1^N\right) \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right) \end{aligned} \tag{4.10}$$

We refer to the probability $p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)$ of observing an image location l at time t given position l_{t-1} and the images X_{t-1}^t as *tracking model*. The tracking model is described in Section 4.3. We further assume that the image feature $X_{R,\,t}^{(1)}$ depends only on the current image location l_t, the current state s_t, and the hypothesized word sequence w_1^N. Thus, Equation (4.10) can be rewritten to:

$$p\left(X_{R,\,t}^{(1)},\, X_t,\, l_t \,|\, s_{t-1}^t,\, l_1^{t-1},\, X_1^{t-1},\, w_1^N\right) = p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, l_t,\, w_1^N\right) \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right) \tag{4.11}$$

The probability $p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right)$ of observing the foreground region of image feature one of image X_t at a given time t, a given state s_t, and the hypothesized word sequence w_1^N is called *visual emission probability*. Using Equations (4.7) and (4.11), the probability of observing an image sequence X_1^T given a word sequence w_1^N is phrased as:

$$p\left(X_1^T \,|\, w_1^N\right) = \sum_{s_1^T} \sum_{l_1^T} \prod_{t=1}^{T} p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right) \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right) \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \tag{4.12}$$

Since we are searching for the word sequence w_1^N maximizing Equation (4.1), it is sufficient to approximate the summations in Equation (4.12) by the corresponding maxima.

$$p\left(X_1^T \,|\, w_1^N\right) \approx \max_{s_1^T} \max_{l_1^T} \prod_{t=1}^{T} p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right) \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right) \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \tag{4.13}$$

To control the influence of the individual models, we apply a state transition scale η to the state transition probability, a visual emission scale δ to the visual emission probability, and a tracking model scale γ to the tracking model.

$$p\left(X_1^T \,|\, w_1^N\right) = \max_{s_1^T} \max_{l_1^T} \prod_{t=1}^{T} p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right)^{\delta} \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)^{\gamma} \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right)^{\eta} \tag{4.14}$$

4.3 Tracking Model

We rely on the dynamic programming approach described in Section 2.2 and derived in [Rybach 06] to define the tracking model.

The logarithm of the tracking model probability $p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)$ can be expressed by a scoring function $\tilde{g}\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$ rating the current object location l_t depending on the preceding object location l_{t-1} and the images X_{t-1}^t. To fulfill the requirements of a probability density function, the scoring function $\tilde{g}$ must be normalized by the sum over all possible object locations for the given time frame.

$$\log\left(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)\right) = \frac{\tilde{g}(l_t,\, l_{t-1},\, X_{t-1}^t)}{\sum\limits_{l'} \tilde{g}(l',\, l_{t-1},\, (X_{t-1}^t)} \tag{4.15}$$

Since the normalization part is constant with respect to a given object location, it can be omitted in the maximization of Equation (4.1).

The scoring function $\tilde{g}\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$ is split into an image independent part $\mathcal{J}(l_t,\, l_{t-1})$ controlling properties of the chosen object path and an image dependent part $g\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$. Analogous to Section 2.2, $\mathcal{J}(l_t,\, l_{t-1})$ is denoted *jump penalty function*. The squared Euclidean point distance is used here as jump penalty

$$\mathcal{J}(l,\, l') = \alpha_{\mathcal{J}} \cdot \parallel l - l' \parallel^2 = \alpha_{\mathcal{J}} \cdot (l - l')^T \cdot (l - l') \tag{4.16}$$

where $\alpha_{\mathcal{J}}$ is a weighting factor. Using these score functions, the logarithm of the tracking model probability can be expressed by:

$$\log\left(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)\right) = g\left(l_t,\, l_{t-1},\, X_{t-1}^t\right) - \mathcal{J}(l_t,\, l_{t-1}) \tag{4.17}$$

The jump penalty is subtracted to penalize wide movements of the object of interest. We assume a constant background and motion history in the tracking model to track an object. Thus, the scoring function $g\left(l_t,\, l_{t-1},\, X_{t-1}^t\right)$ is defined by Equation (2.12):

$$\begin{aligned} -\, g(l_t,\, l_{t-1},\, X_{t-1}^t) \;&=\; \alpha_1 \sum_{l \in Q_t} \left(X_t[l_{t-1} + l] + X_{t-1}[l_{t-1} + l]\right)^2 \\ &+\; \alpha_2 \sum_{l \notin Q_t \cup Q_{t-1}} \left(X_t'[l]\right)^2 \end{aligned} \tag{4.18}$$

4.4 Optimization Criterion

Using Equation (4.14), the language model scale α, and the visual model scale β, the optimization problem in Equation (4.1) of joint tracking and recognition is formulated as:

$$\begin{aligned} [w_1^N]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \; \Bigg\{ & p\left(w_1^N\right)^{\alpha} \cdot \\ & \left(\max_{[s_1^T,\, l_1^T]} \prod_{t=1}^{T} p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right)^{\delta} \cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)^{\gamma} \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right)^{\eta} \right)^{\beta} \Bigg\} \end{aligned} \tag{4.19}$$

In the following, we omit all scales except the tracking model scale γ for readability. Since we intend to solve the optimization problem using dynamic programming, Equation (4.19) is rephrased in terms of scores by applying the negative logarithm to all probabilities.

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\operatorname{argmin}} \left\{ -\log(p\left(w_1^N\right)) + \min_{[s_1^T, l_1^T]} \sum_{t=1}^{T} -\log\left(p\left(X_{R,t}^{(1)} \,|\, s_t,\, w_1^N\right)\right) \right.$$
$$\left. -\gamma \cdot \log\left(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)\right) - \log(p\left(s_t \,|\, s_{t-1},\, w_1^N\right)) \right\} \tag{4.20}$$

Assuming an unigram language model and changing from lattices to single word models, Equation (4.20) is transformed to (4.22).

$$[w_1^N]_{\text{opt}} =$$
$$\underset{w_1^N,\, N}{\operatorname{argmin}} \left\{ \sum_{n=1}^{N} -\log(p\left(w_n\right)) + \min_{[s_1^T, l_1^T]} \sum_{t=1}^{T} -\log\left(p\left(X_{R,t}^{(1)} \,|\, s_t,\, w_1^N\right)\right) \right.$$
$$\left. -\gamma \cdot \log\left(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)\right) - \log(p\left(s_t \,|\, s_{t-1},\, w_1^N\right)) \right\} \tag{4.21}$$
$$= \underset{w_1^N,\, N,\, t_1^N}{\operatorname{argmin}} \left\{ \sum_{n=1}^{N} \left[-\log(p\left(w_n\right)) + \min_{s_{t_{n-1}+1}^{t_n}} \min_{l_{t_{n-1}+1}^{t_n}} \sum_{t=t_{n-1}+1}^{t_n} -\log\left(p\left(X_{R,t}^{(1)} \,|\, s_t,\, w_n\right)\right) \right.\right.$$
$$\left.\left. -\gamma \cdot \log\left(p\left(l_t \,|\, l_{t-1},\, X_{t-1}^t\right)\right) - \log(p\left(s_t \,|\, s_{t-1},\, w_n\right)) \right] \right\} \tag{4.22}$$

Substituting $-\log(p\left(s_t \,|\, s_{t-1},\, w_n\right))$ with the TDP (Cf. Section 2.3) and unfolding the tracking model in Equation (4.17) yields

$$[w_1^N]_{\text{opt}} =$$
$$\underset{w_1^N,\, N,\, t_1^N}{\operatorname{argmin}} \left\{ \sum_{n=1}^{N} \left[-\log(p\left(w_n\right)) + \min_{s_{t_{n-1}+1}^{t_n}} \min_{l_{t_{n-1}+1}^{t_n}} \sum_{t=t_{n-1}+1}^{t_n} -\log\left(p\left(X_{R,t}^{(1)} \,|\, s_t,\, w_n\right)\right) \right.\right.$$
$$\left.\left. -\gamma \cdot \left[g\left(l_t,\, l_{t-1},\, X_{t-1}^t\right) - \mathcal{J}(l_t,\, l_{t-1}) \right] + \mathcal{T}\left(s_t - s_{t-1},\, w_n\right) \right] \right\} \tag{4.23}$$
$$= \underset{s_1^T, l_1^T, w_1^T}{\operatorname{argmin}} \left\{ \sum_{t=1}^{T} \tilde{q}\left(X_{t-1}^t,\, s_t,\, l_t \,|\, s_{t-1},\, l_{t-1},\, w_t\right) \right\} \tag{4.24}$$

for a suitable definition of the auxiliary function $\tilde{q}\left(X_{t-1}^t,\, s_t,\, l_t \,|\, s_{t-1},\, l_{t-1},\, w_t\right)$.

The optimization criterion (4.23) needs to be evaluated at within word models and at word boundaries. Hence, we define the auxiliary function as follows:

within words:

$$\tilde{q}\left(X_{t-1}^{t},\, s,\, l \,|\, s',\, l',\, w_t\right) = -\log\left(p\left(X_{R,t}^{(1)} \,|\, s,\, w_t\right)\right) - \gamma \cdot \left[g\left(l,\, l',\, X_{t-1}^{t}\right) - \mathcal{J}(l,\, l')\right] + \mathcal{T}\left(s - s',\, w_t\right) \tag{4.25}$$

word boundaries:

$$\tilde{q}\left(X_{t-1}^{t},\, s,\, l \,|\, s',\, l',\, w_t\right) = -\log p\left(w_t\right) + \tilde{q}\left(X_{t-1}^{t},\, s,\, l \,|\, s' \,=\, 0,\, l',\, w_t\right) \tag{4.26}$$

where $s' = 0$ is the virtual starting state for each word HMM.

4.5 Dynamic Programming Recursion

In the following we derive the dynamic programming recursion for the joint optimization problem of Section 4.4. We define the distance table $D\left(t,\, s,\, l;\, w\right)$ of the best partial path up to time t ending in state s of word w for the object of interest at image location l as

$$D\left(t,\, s,\, l;\, w\right) = \min_{[s_1^T,\, l_1^T,\, w_1^T]} \left\{ \sum_{\tau=1}^{t} d\left(X_{\tau-1}^{\tau},\, s_\tau,\, l_\tau \,|\, s_{\tau-1},\, l_{\tau-1},\, w_\tau\right) : \left(s_t,\, l_t,\, w_t\right) = \left(s,\, l,\, w\right) \right\} \tag{4.27}$$

Since the tracking model (Cf. Section 4.3) is independent of the trained word models, we calculate a single tracking model table $V(t,\, l)$

$$V(t,\, l) \;=\; \min_{l' \in \mathcal{M}(l)} \left\{ V(t-1,\, l') + v(t,\, l,\, l') \right\} \tag{4.28}$$

$$v(t,\, l,\, l') \;=\; \mathcal{J}(l,\, l') - g\left(l,\, l',\, X_{t-1}^{t}\right) \tag{4.29}$$

where $\mathcal{M}(l)$ is the set of possible predecessor locations l' for an image location l. Using the tracking model table V, the recursion equations for the two distinct transition situations are given as

within words:

$$D\left(t,\, s,\, l;\, w\right) = \min_{s',\, l'} \{ D\left(t-1,\, s',\, l';\, w\right) + d\left(X_{t-1}^{t}, s,\, l \,|\, s',\, l',\, w\right) \} + \gamma \cdot V(t,\, l) \tag{4.30}$$

word boundaries:

$$D\left(t,\, s = 0,\, l;\, w\right) = -\log p\left(w\right) + \min_{k,v} \{ D\left(t,\, S\left(v\right), k;\, v\right) \} \tag{4.31}$$

where the local distance $d\,(X_{t-1}^{t},\, s,\, l \,|\, s',\, l',\, w)$ is given by

$$d\,(X_{t-1}^{t},\, s,\, l \,|\, s',\, l',\, w) = -\log\left(p\left(X_{R,t}^{(1)} \,|\, s,\, w_t\right)\right) + \mathcal{T}(s - s',\, w) \tag{4.32}$$

Special care must be taken to ensure that the function encoded in the distance score table D is monotonic increasing. Hence, all components must be normalized to a common value range and must be greater or equal to zero.

To keep track of the chosen path, a backpointer table $B(t,\, s,\, l;\, w)$ is filled analogous to the distance score table by replacing min by argmin in Equations (4.27), (4.30) and (4.31).

4.6 Complexity

In the following let T be the average number of time frames of a sentence, W be the number of reference words, S the average number of states for all reference words, and L the average number of active locations in an image.

We first discuss the time complexity of the reference system as described in Chapter 2 and compare it to the time complexity of the proposed integrated tracking and recognition system. In the reference system tracking is part of the preprocessing. In the proposed integrated tracking and recognition system, tracking is part of the recognition process. After the discussion of the time complexities of both systems, the space complexities of both systems are presented and compared.

4.6.1 Time Complexity

We focus on the discussion of the time complexity of visual search in the reference system and the proposed integrated system because both systems perform the same number of comparisons in the language model network. Visual search describes the process of unfolding the visual models during the recognition phase.

Reference System

The reference systems applies tracking in the feature extraction phase. The tracking framework needs to evaluate for each active image location all possible image locations in the preceding image as possible predecessor locations. Thus, the tracking framework performs L^2 comparisons per image. Assuming constant time to evaluate the tracking scoring functions, the time complexity of the tracking framework is $\mathcal{O}(TL^2)$.

Furthermore, the reference system must evaluate all trained reference word models at each time frame t during recognition. Assuming HMMs with Bakis topology and a one-state silence model, the number comparisons needed in the visual model is $\mathcal{O}(3 \cdot T \cdot [WS + 1])$ for a unigram language model, $\mathcal{O}(3 \cdot T \cdot [WS + W])$ for a bigram

language model, and $\mathcal{O}(3 \cdot T \cdot W \cdot [WS + 1])$ for a trigram language model. Assuming trigram language model for the reference system, the overall time complexity is given by

$$\mathcal{O}(TL^2) + \mathcal{O}(3 \cdot T \cdot W \cdot [WS + 1]) = \mathcal{O}(T \cdot [L^2 + 3 \cdot W \cdot [WS + 1]]) \tag{4.33}$$

which is feasible for real-world image sizes such as 195 by 165 pixel.

Integrated Tracking and Recognition

The proposed integrated tracking and recognition system uses no distinct feature extraction phase. Conversely, feature extraction is part of the recognition phase. As with the tracking framework of the reference system, the integrated tracking and recognition system needs to calculate the tracking model table V at each time frame t. Starting from the preceding time frame $t-1$, all active candidate object locations must expanded in time reaching again all possible active object locations at time t. Assuming that unnecessary calculations are prevented, the overall time complexity to calculate the tracking model table V for T frames is $\mathcal{O}(TL^2)$. Moreover, the integrated system must expand each state-location tuple of every trained reference word in time (state space) and spatial domain as shown in Figure 4.2b (Page 25). Assuming Bakis topology, the integrated system must evaluate $3 \cdot L$ possible preceding state-location tuples at time $t-1$ for each tuple at time t. Thus, the overall time complexity of visual search in the proposed integrated tracking and recognition system is $\mathcal{O}(T \cdot [\,3 \cdot L^2 \cdot [WS + 1] + L^2]\,)$ for an unigram language model, $\mathcal{O}(T \cdot [\,3 \cdot L^2 \cdot [WS + W] + L^2]\,)$ for a bigram language model, and $\mathcal{O}(T \cdot [\,3 \cdot L^2 \cdot W \cdot [WS + W] + L^2]\,)$ for a trigram language model. The time complexities of the integrated system are subsumed in Table 4.1.

Comparing the overall time complexity of the integrated tracking and recognition system to the overall time complexity of the reference system, i.e. Equation (4.33), the time complexity of the proposed integrated system is much higher than the time complexity of the reference system. Considering again real world image sizes of 195 by 165 pixel, combined linear search is not feasible and suitable pruning techniques must be applied.

4.6.2 Space Complexity

In contrast to time complexity, we need to consider both visual model and language model complexities to evaluate the total space complexity of the proposed integrated tracking and recognition system and the reference system.

Table 4.1. Time Complexity of Combined Linear Search

	visual search	language model
unigram	$T \cdot [3 \cdot L^2 \cdot [W \cdot S + 1] + L^2]$	$T \cdot [W + 1]$
bigram	$T \cdot [3 \cdot L^2 \cdot [W \cdot S + W] + L^2]$	$T \cdot W \cdot [W + 1]$
trigram	$T \cdot [3 \cdot L^2 \cdot W \cdot [W \cdot S + W] + L^2]$	$T \cdot W \cdot [W^2 + 1]$

Reference System

In the visual search, the reference system needs to store a hypothesis of each state for every reference word and the corresponding backpointer. Hence, the reference system has a space complexity in visual search of $\mathcal{O}(2 \cdot [WS + 1])$ for an unigram language model, $\mathcal{O}(2 \cdot [WS + W])$ for a bigram language model, and $\mathcal{O}(2 \cdot W \cdot [WS + W])$ for the trigram language model.

In the language model search, the reference system needs to store a language model network node and its predecessor for every time frame. Hence, the space complexities for the language model search in the reference system are $\mathcal{O}(2 \cdot T)$ in the unigram case, $\mathcal{O}(2 \cdot T \cdot [W + 1])$ in the bigram case, and $\mathcal{O}(2 \cdot T \cdot [W^2 + 1])$ in the trigram case.

Integrated Tracking and Recognition

In the visual search, the integrated tracking and recognition systems needs to store the tracking model table of size L, a hypothesis for each state-location tuple for every reference word and the corresponding backpointers at every time frame. Thus, the overall space complexity of visual search in integrated tracking and recognition is $\mathcal{O}(2 \cdot L \cdot [W \cdot S + 1] + L)$ for an unigram language model, $\mathcal{O}(2 \cdot L \cdot [W \cdot S + W] + L)$ for a bigram language model, and $\mathcal{O}(2 \cdot L \cdot W \cdot [W \cdot S + W] + L)$ for a trigram language model.

In the language model search, the integrated tracking and recognition system stores the language model node, a backpointer and the hypothesized object location at every time frame. Thus, the integrated tracking and recognition has a space complexity of $\mathcal{O}(3 \cdot T)$ in the unigram case, $\mathcal{O}(3 \cdot T \cdot [W + 1])$ in the bigram case, and $\mathcal{O}(3 \cdot T \cdot [W^2 + 1])$ in the trigram case. An overview on the space complexities of the proposed integrated tracking and recognition system is presented in Table 4.2.

In comparison to the reference system and in contrast to the time complexities, space complexity in visual search for combined linear search is feasible.

Table 4.2. Space Complexity of Combined Linear Search

	visual search	language model
unigram	$2 \cdot L \cdot [W \cdot S + 1] + L$	$3 \cdot T$
bigram	$2 \cdot L \cdot [W \cdot S + W] + L$	$3 \cdot T \cdot [W + 1]$
trigram	$2 \cdot L \cdot W \cdot [W \cdot S + W] + L$	$3 \cdot T \cdot [W^2 + 1]$

4.7 Pruning Criteria

The visual search space of the proposed integrated tracking and recognition system is very large because the system combines temporal and spatial information (Cf. Section 4.6). Hence, the system is infeasible if the search space is not reduced. The integrated tracking and recognition system uses value thresholding on three distinct score levels to reduce the visual search space. Pruning can be performed on the *tracking model level*, *visual emission level*, and *visual model level*.

4.7.1 Tracking Model Level

The tracking model is modeled to be independent from the trained word models. Therefore, pruning of candidate object locations effects all state-location hypotheses of all trained reference words. Pruning on the tracking model level only prunes the spatial domain.

Given a fixed value threshold f_{TR} for the tracking model and the minimal value

$$\mathcal{V}_{TR}(t) := \min_{l}\{V(t,\, l)\} \tag{4.34}$$

a state-location hypothesis $(t,\, s,\, l;\, w)$ of a word w is pruned at time t iff

$$V(t,\, l) > f_{TR} \;+\; \mathcal{V}_{TR}\,(t). \tag{4.35}$$

4.7.2 Visual Emission Level

On the visual emission level, pruning is performed on the HMM state level only. The tracking model score $V(t,\, l)$ is part of the accumulated score in the score table D as defined by Equations (4.30), (4.31), and (4.32). Since we prune only at the visual emission level, the tracking model score must be removed from the accumulated score table.

We define $D_{VS}(t,\, s,\, l;\, w)$ as the projection of the distance table $D(t,\, s,\, l;\, w)$ on the visual emission model, jump penalty, and language model scores, i.e.

$$D_{VS}(t,\, s,\, l;\, w) = D(t,\, s,\, l;\, w) - V(t,\, l) \tag{4.36}$$

Given a fixed value threshold f_{VS} and the minimal value

$$\mathcal{D}_{VS}(t) := \min_{w,s,l}\{D_{VS}(t, s, l; w)\} \tag{4.37}$$

a state-location hypothesis $(t, s, l; w)$ of a word w is pruned at time t iff

$$D_{VS}(t, s, l; w) > f_{VS} + \mathcal{D}_{VS}(t). \tag{4.38}$$

4.7.3 Visual Model Level

Pruning on the visual model level comprises the state space and the spatial domain. In contrast to pruning on the two levels described before, pruning on the visual model level is performed using the integrated tracking and recognition score accumulated in score table D. Thus, pruning on the visual model level is performed as follows:

Given a fixed value threshold f_H and the minimal value

$$\mathcal{D}_H(t) := \min_{w,s,l}\{D(t, s, l; w)\} \tag{4.39}$$

a state-location hypothesis $(t, s, l; w)$ is pruned on hypothesis level iff

$$D(t, s, l; w) > f_H + \mathcal{D}_H(t). \tag{4.40}$$

4.8 Overview of the Combined System

The proposed integrated tracking and recognitions system solves a combined optimization problem for tracking and recognition. In contrast to the reference system, tracking is part of the recognition phase and not performed in a distinct feature extraction phase. Therefore, feature extraction is performed implicitly in the integrated recognition phase of the proposed integrated tracking and recognition system. The structure of the integrated tracking and recognition system is depicted in Figure 4.1.

Moreover, tracking decisions reached in integrated tracking and recognition are influenced by the trained object models leading to an overall tracking path optimized according to the hypothesized word sequence. Hence, the proposed integrated system adapts features to the trained object models.

Unfortunately, the proposed tracking and recognition system is infeasible for real world images sizes due to the high time complexity in visual search (Cf. Section 4.6). Therefore, pruning must be applied to reduce the large visual search space possibly introducing recognition errors.

Chapter 5

Rescoring and Model-Driven Tracking Adaptation

In this chapter we propose efficient approximations to the proposed integrated tracking and recognition system. The time complexity of integrated tracking and recognition is very high in visual search. Still, we want to integrate tracking and recognition and to additionally optimize the tracking path w.r.t. the hypothesized word sequence. Since the two-phase reference system of Chapter 2 is computationally feasible, approximations to the integrated tracking and recognition system are implemented in the reference system.

The proposed approximations are divided into rescoring and model-driven tracking path adaptation approaches.

In rescoring several distinct tracking paths are extracted in the feature extraction phase of the reference system. Recognition is then performed for each of these paths independently.

In contrast to rescoring, model-driven tracking path adaptation uses only the best tracking path according to the chosen tracking criterion and adapts this path w.r.t. the trained visual models in the recognition phase.

Both approaches break the strict two-phase concept of the reference system. Feature extraction in the recognition phase is performed in addition to the distinct feature extraction phase.

5.1 Rescoring

The dynamic programming framework described in Section 2.2 extracts tracking paths according to a beforehand specified tracking criterion. The reference system uses only the best path i.e. the global optimum w.r.t. the chosen tracking criterion in the extraction of tracking dependent features. The best tracking path might contain errors leading to errors in recognition although the best path is optimal w.r.t. the chosen tracking criterion. The remaining m-best tracking paths extracted in feature extraction are not optimal w.r.t. the chosen tracking criterion. Still, one or more paths

Figure 5.1. Visualization of the 450 best tracking paths w.r.t. motion tracking criterion. Best path marked in yellow.

among the m-best paths might track the object to be tracked more accurately than the globally optimal path w.r.t. the tracking criterion. An example of this effect is depicted in Figure 5.1 with the globally optimal tracking path marked by the yellow rectangle. In the first frame the dominant hand is tracked correctly but in succeeding frames the best path tracks the non-dominant hand while other paths still correctly follow the dominant hand.

The dynamic programming tracking framework of Section 2.2 extracts the globally optimal tracking path w.r.t. the chosen tracking criterion by tracing back tracking decisions starting from the best hypothesized tracking location in the final frame of the input video. Multiple tracebacks, i.e. tracking paths, can be extracted by tracing back starting from globally not optimal tracking locations in the final frame of the input video. Thus, a sorted list $l^T_{1,1}, \ldots, l^T_{1,m}$ of the m-best tracking paths is extracted according to Equations (5.1) and (5.2) from a video sequence.

$$l_{T,i} = \underset{(x,y)\notin\{l_{T,1},\ldots,l_{T,i-1}\}}{\operatorname{argmax}} \{S(T, x, y)\} \tag{5.1}$$

$$l_{t-1,i} = B(t, l_{t-1,i}), \qquad \text{for } i = 1, \ldots, m \tag{5.2}$$

Figure 5.2 schematically shows the perfect oracle tracking path in red, the globally optimal tracking path in blue and a list of M tracking paths with the m-th path being depicted as the dashed green curve. The upper part of the figure shows the oracle path and the spatial positioning of the globally best tracking path, the m-th best, and the M-th best tracking to the oracle path. Although the globally best tracking path fits the oracle path over the complete depicted sequence best, the distance of the m-th and M-th best tracking paths to the oracle path is lower than the distance of the globally best path to the oracle path at single frames. Hence, there are two possibilities to perform rescoring using a m-best tracking path list.

First, m-best path rescoring (Cf. Section 5.1.1) performs recognition for every path $l^T_{1,i}$ independently. The system's overall recognition result is the best result obtained from the single rescoring recognitions. The process of m-best rescoring is depicted in

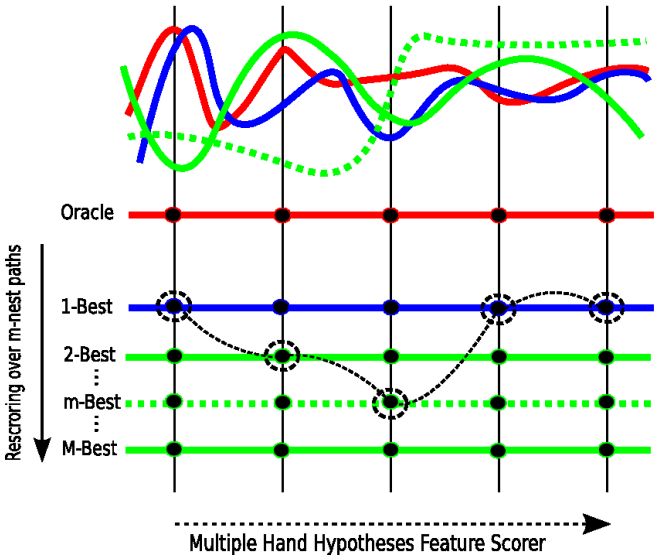


Figure 5.2. Overview m-best path rescoring vs. Multiple Hand Hypotheses

Figure 5.2 by choosing and testing the extracted paths of the m-best tracking path list from top to bottom.

Second, we perform recognition on all tracking paths in the m-best tracking path list simultaneously in the multiple hand hypotheses approach and choose the best fitting tracking location among the candidate locations w.r.t. the trained word models at every time frame. Thus, a new tracking path is generated from the m-best tracking list. The process of multiple hand hypotheses is shown in Figure 5.2 by choosing candidate locations (marked by dashed black circles) from left to right. Hence, the spatial context encoded in the tracking paths is discarded.

Furthermore, spatial and temporal context can be integrated in the multiple hand hypotheses approach by taking the tracking scores of the distinct candidate tracking locations into account. This extended multiple hand hypotheses approach is equivalent to the proposed integrated tracking and recognition approach of Chapter 4 with reduced search space. The extended multiple hand hypotheses approach is discussed in Section 5.1.3.

5.1.1 m-best Tracking Path Rescoring

In m-best tracking path rescoring recognition is performed for each path in the m-best list independently. The overall recognition result of the m-best tracking path rescoring system is the best recognition result obtained on the independent m-best paths.

Figures 5.1 and 5.2 show that correct tracking candidate locations might be present in the suboptimal tracking paths of the m-best tracking path list even if the globally optimal tracking path tracks the wrong object at the given time frame. Therefore, a suboptimal tracking path w.r.t. the chosen tracking criterion might lead to a better feature sequence reducing the number of recognition errors.

Given a list of the m-best tracking paths $l_{1,1}^T, \ldots, l_{1,m}^T$, the optimization problem of m-best tracking path rescoring is phrased as follows:

$$
\begin{aligned}
&[w_1^N]_{\text{opt}} \\
&= \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(w_1^N\right) \cdot \max_{[i : l_1^T := (l_{1,\,i}, \ldots, l_{T,\,i})]} \left\{ p\left(f(X_1^T,\, l_1^T) \,|\, w_1^N\right) \right\} \right\} \qquad (5.3) \\
&= \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(w_1^N\right) \cdot \right. \\
&\qquad \left. \max_{[s_1^T]} \max_{[i : l_1^T := (l_{1,\,i}, \ldots, l_{T,\,i})]} \prod_{t=1}^{T} \left\{ p\left(f(X_t,\, l_t) \,|\, s_t,\, w_1^N\right) \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\} \right\} \qquad (5.4)
\end{aligned}
$$

where $f(X,\, l)$ is a feature function extracting a feature from image X in the vicinity of location l. By incorporating multiple different tracking paths into recognition, it is possible to recover from tracking errors as long as the target is among the given candidate locations present in the m-best tracking paths. Recovery from tracking errors is possible even where the preprocessing failed miserably such as confusing dominant and non-dominant hand. By using complete tracking paths including the globally best tracking path, m-best tracking path rescoring is guaranteed to achieve the recognition performance of the reference system.

In speech recognition a approach similar to m-best tracking path rescoring is known as *acoustic rescoring* [Jelinek 98].

5.1.2 Multiple Hand Hypotheses

In the multiple hand hypotheses approach, we perform recognition on all tracking paths in the m-best tracking path list simultaneously. In contrast to the proposed m-best tracking path rescoring of Section 5.1.1, the multiple hand hypotheses approach selects a best fitting candidate object location from all tracking paths in the m-best tracking path list at each time frame t and not a complete tracking path from the m-best list. Therefore, the spatial context of the tracking paths is explicitly discarded.

Figure 5.1 shows that candidate locations present in the m-best tracking path list can correctly track the dominant hand while the candidate locations of the globally best tracking path (marked by the yellow rectangle) w.r.t. the chosen tracking criterion incorrectly tracks the non-dominant hand. According to Figure 5.2, globally not

optimal tracking paths can provide better candidate locations than the globally optimal tracking path for single frames. The multiple hand hypotheses approach is able to correct tracking errors by locally choosing a candidate location from all tracking paths in the m-best list as long as the correct location is among the given candidate locations. The lower half of Figure 5.2 depicts the process of locally choosing candidate locations from the paths in the m-best tracking path list by choosing the black dotted path from left to right.

In the recognition phase of the reference system of Chapter 2 the multiple hand hypotheses approach considers and selects a candidate location from the set of candidate locations $\{l_{t,1}, \ldots, l_{t,m}\}$ depending on the hypothesized word sequence w_1^N at each time frame t. Thus, the decision rule of the multiple hand hypotheses approach is phrased as follows:

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(w_1^N\right) \cdot \max_{[s_1^T]} \prod_{t=1}^{T} \max_{i=1,\ldots,m} \left\{ p\left(f(X_t,\, l_{t,i}) \,|\, s_t,\, w_1^N\right) \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\} \right\} \quad (5.5)$$

Using the decision rule in Equation (5.5), the recognition system recombines a tracking path optimal w.r.t. the hypothesized word sequence. The resulting tracking path possible contains candidate locations from different tracking paths as depicted by the black dotted path in Figure 5.2. Therefore, the resulting tracking path might neither be smooth nor present in the m-best tracking path list. Moreover, the spatial and hence temporal context of the candidate locations is discarded by taking local tracking decisions.

Similar to the proposed m-best tracking path rescoring, the multiple hand hypotheses approach is guaranteed to achieve the performance of the reference system because the recombined tracking path may be identical with the globally best tracking path.

5.1.3 Extended Multiple Hand Hypotheses

In the extended multiple hand hypotheses approach we approximate the proposed integrated tracking and recognition system of Chapter 4 by taking the temporal and spatial context of the candidate tracking locations into account in the multiple hand hypotheses approach.

Given a m-best tracking path list, a candidate location $l_{t,i}$ of path i in the m-best tracking path list depends on all candidate locations $l_{1,i}, \ldots, l_{t-1,i}$ of previous time frames. Hence, every candidate location has a distinct temporal and spatial context. The multiple hand hypotheses approach explicitly discards the temporal and spatial context of a candidate location by taking only local tracking decisions. The m-best

tracking path list is sorted according to the overall tracking score achieved by the distinct paths. Figure 5.2 shows that paths with better, i.e. higher tracking scores, have a lower overall distance to the perfect oracle tracking path than tracking paths with an lower overall tracking score. Hence, candidate locations $l_{t,\,i}$ supplied by tracking paths with high overall tracking score are more probable to correctly describe the position of the object to be tracked than candidate locations supplied by tracking paths with lower overall score. Therefore, the temporal and spatial context of a candidate location $l_{t,\,i}$ of path i at time t is modeled by the overall tracking score of path i reaching location $l_{t,\,i}$ at time t. The candidate locations $\{l_{t,\,1},\,\ldots\,,\,l_{t,\,m}\}$ are weighted with their temporal and spatial context. By incorporating the partial tracking scores of the candidate locations into the multiple hand hypotheses approach, the later becomes equivalent to the proposed integrated tracking and recognition system with a fixed search space.

We derive the extended multiple hand hypotheses approach from the integrated tracking and recognition approach detailed in Chapter 4. The decision problem of integrated tracking and recognition has been derived in Equation (4.19) as

$$
\begin{aligned}
[w_1^N]_{\text{opt}} = \underset{w_1^N}{\operatorname{argmax}} \; \Bigg\{ & \, p\left(w_1^N\right) \cdot \\
\max_{[s_1^T,\, l_1^T]} \prod_{t\,=\,1}^{T} & \, p\left(X_{R,\,t}^{(1)} \,|\, s_t,\, w_1^N\right) \;\cdot\; p\left(l_t \,|\, l_{t-1},\, X_{t-1}^{t}\right)^{\gamma} \cdot \; p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \Bigg\}
\end{aligned}
\tag{5.6}
$$

omitting all scale factors except the tracking model scale γ for readability. For convenience, we define the feature extraction function f as follows:

$$
f(X, l) = X_R^{(1)} \tag{5.7}
$$

The extended multiple hand hypotheses approach restricts the overall search space to a set of object candidate locations $l_{t,1},\,\ldots\,,l_{t,m}$ present in the m-best tracking paths from the feature extraction phase at each time frame t. Assuming such a restriction to beforehand extracted paths, the dependency of the tracking model on the images X_{t-1}^{t} is dropped. Thus, the tracking model used is

$$
p\left(l_t \,|\, l_{t-1},\, X_{t-1}^{t}\right) = p\left(l_{t,\,i} \,|\, l_{t-1,\,i}\right) \tag{5.8}
$$

given a tracking path i. Using Equations (5.7) and (5.8) and maximizing over m-best tracking paths instead of all possible object locations in an image, Equation (5.6) is

rewritten to

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{ p\left(w_1^N\right) \cdot \max_{[s_1^T]} \left\{ \prod_{t=1}^{T} \max_{i=1,,\dots,m} \left\{ p\left(f(X_t,\, l_{t,i}) \,|\, s_t,\, w_1^N\right) \;\cdot\; p\left(l_{t,i} \,|\, l_{t-1,i}\right)^{\gamma} \cdot\; p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\} \right\} \right\} \tag{5.9}$$

Applying the negative logarithm to Equation (5.9), the optimization problem is rewritten in terms of scores.

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\text{argmin}} \left\{ -\log p\left(w_1^N\right) + \min_{[s_1^T]} \sum_{t=1}^{T} \min_{i=1,,\dots,m} \left\{ -\log p\left(f(X_t,\, l_{t,i}) \,|\, s_t,\, w_1^N\right) - \gamma \cdot \log p\left(l_{t,i} \,|\, l_{t-1,i}\right) - \log p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\} \right\} \tag{5.10}$$

$$= \underset{[s_1^T],\,[w_1^T]}{\text{argmin}} \left\{ \sum_{t=1}^{T} \min_{i=1,,\dots,m} \left\{ \tilde{q}\left(X_t,\, s_t,\, l_{t,i} \,|\, s_{t-1},\, l_{t-1,i},\, w_t\right) \right\} \right\} \tag{5.11}$$

The auxiliary function $\tilde{q}$ is defined analogously to the auxiliary function derived for the integrated tracking and recognition approach (Cf. Section 4.4). Using dynamic programming, Equation (5.11) is solved by calculating a distance table D for each partial recognition path ending at time t in state s for each reference word w.

$$D(t,\, s;\, w) = \min_{[s_1^T,\, w_1^T]} \left\{ \sum_{\tau=1}^{t} \min_{i=1,\dots,m} \left\{ d\left(X_\tau,\, s_\tau,\, l_{\tau,i} \,|\, l_{\tau-1,i},\, s_{\tau-1},\, w_\tau\right) \right\} \right\} \tag{5.12}$$

Assuming a unigram language model, the recursive equations are derived for the within word case as

$$D(t,\, s;\, w) = \min_{s'} \min_{i=1,\dots,m} \left\{ D(t-1,\, s';\, w) + d\left(X_t,\, s,\, l_{t,i} \,|\, s',\, l_{t-1,i},\, w\right) \right\} \tag{5.13}$$

and at word boundaries as

$$D(t,\, s=0;\, w) = -\log p\left(w\right) + \min_{v} \left\{ D(t,\, S(v);\, v) \right\} \tag{5.14}$$

where

$$d\left(X_t,\, s,\, l_{t,i} \,|\, s',\, l_{t-1,i},\, w\right) = -\log p\left(f(X_t,\, l_{t,i}) \,|\, s,\, w\right) - \gamma \cdot \varphi(t,\, l_{t,i},\, l_{t-1,i}) + \mathcal{T}(s - s',\, w)\ . \tag{5.15}$$

Due to the restriction of the spatial search space to a set of tracking paths extracted beforehand, and the auxiliary function φ being independent of the current word hypothesis w and the current state s, φ can be defined as

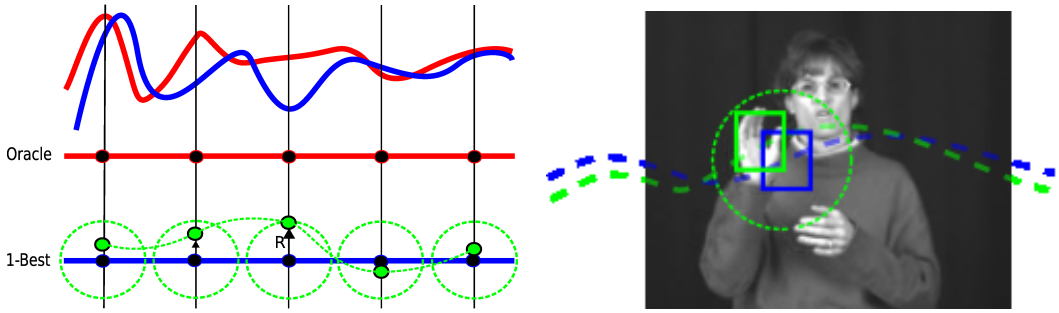
$$\varphi(t,\, l_{t,i},\, l_{t-1,i}) = \varphi(t,\, l_{t,i}) = \log\left(1 - \frac{S(t,\, l_{t,i})}{\sum_{i'=1}^{m} S(t,\, l_{t,i'})}\right) \tag{5.16}$$

where $S(t,\, l_{t,i})$ is the tracking score of the best partial tracking path i up to time t ending in location l (Cf. Section 2.2). Since the extended multiple hand hypotheses approach uses the tracking scores calculated during the feature extraction phase, tracking scores $S(t,\, l_{t,i})$ of partial rescoring paths need to be renormalized to fulfill the requirements of a probability density function. The tracking framework presented in Section 2.2 maximizes scores while integrated rescoring minimizes scores. Therefore, the renormalized tracking scores are interpreted as one minus the probability of reaching location $l_{t,i}$ at time t on path i.

5.2 Model-driven Tracking Adaptation

In model-driven tracking adaptation, we propose to locally adapt a given tracking path w.r.t. the hypothesized word sequence. The tracking framework of Section 2.2 extracts a globally optimal or single-best tracking path according to a beforehand chosen tracking criterion. The tracking criterion is independent of the trained word models. Moreover, the reference system depends on accurate tracking in feature extraction because appearance-based features shifted by only a few pixel lead to different scores in recognition. Tracking errors lead to recognition errors which normally can not be compensated for in the recognition phase. Although the single-best tracking path is assumed to lead to the best possible feature sequence, the single-best tracking path still contains tracking errors (Cf. Figure 5.3b). Assuming that the single-best tracking path (blue path in Figure 5.3a) is close to the perfect oracle tracking path (red path in Figure 5.3a), we want to recover tracking errors up to a certain range R. The adaptation principle is shown in the lower half of Figure 5.3a. Starting from the object location proposed by the blue single-best path at a time frame t, we extract features at all locations in the green circle region of range R and choose among these features the best feature and location according to the currently hypothesized word sequence. Thus, the dotted green path in Figure 5.3a is a model-driven adapted version of the original single-best tracking path.

Figure 5.3b shows the proposed adaptation technique for a single frame of the RWTH-BOSTON-104 database (Cf Section 6.1). The single-best tracking path is again depicted by the blue path and the adapted path by the green path. Inside the green circle features are extracted at each possible location. The image inside the blue rectangle has been extracted according to the single-best tracking path and the image



(a): Adaptation Principle **(b):** Single Frame Adaptation

Figure 5.3. Model-Driven Tracking Adaptation Principle

inside the green rectangle according to an adaptation location. For each state s of a reference word w, the proposed model-driven tracking adaptation selects the feature with the smallest distance to the trained visual model.

It is not known a-priori in what direction from the single-best tracking path the perfect oracle path is situated. Thus, the choice of the range R is critical. If the range R is too large the green circle in Figure 5.3b includes many locations belonging to the background possibly perfect fitting the trained silence model. If the range R is too small it is impossible to correct a tracking error.

Let the *adaptation range* R be the maximally allowed adaptation distance in pixels from the single-best tracking path as shown in Figure 5.3a. Let further $\delta \in \{(x, y) : -R \leq x \leq R, -R \leq y \leq R\}$ be an adaptation point and let $f(X, l + \delta)$ be a hand feature extracted from image X at location $l+\delta$. The optimization problem for model-driven tracking adaptation is defined by

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\text{argmax}} \left\{ p\left(w_1^N\right) \cdot \right.$$
$$\left. \max_{[s_1^T]} \prod_{t=1}^{T} \max_{\substack{\delta \in \{(x,y):\\ -R \leq x \leq R, -R \leq y \leq R\}}} \left\{ p\left(f(X_t,\, l_t + \delta) \,|\, s_t,\, w_1^N\right) \right\} \cdot p\left(s_t \,|\, s_{t-1},\, w_1^N\right) \right\}. \tag{5.17}$$

The proposed model-driven tracking adaptation approach is limited by the manually chosen adaptation range R. Assuming e.g. a tracking range of five pixel, the system is not able to correct a tracking error at a given time frame if the proposed single best tracking location at time t differs by more than five pixel from the correct hand location.

Model-driven tracking adaptation adapts the given single-best tracking path by distorting it w.r.t. the trained visual models. Feature distortion and feature sampling

are well-known concepts in particle filtering [Arulampalam & Maskell$^+$ 02].

As with the multiple hand hypotheses and extended hand hypotheses approaches, the resulting adaptation path can not be guaranteed to be smooth because the system makes local decisions at each time frame t. However, the object to be tracked is unlikely to move a great distance between consecutive frames. Therefore, the resulting adapted tracking paths should be smooth even without applying explicit smoothness constraints during adaptation.

5.2.1 Distance Weighting

In distance weighting, we assume the single-best tracking path to be close to the perfect oracle path at all time frames. Hence, small adaptation distortions from the single-best tracking path probably lead to better feature sequences than large distortions.

Figure 5.1 shows that the assumption of the single-best tracking path being close to the oracle path i.e. the location of the dominant hand is not always valid. Therefore, we still allow large distortions in adaptation but penalize them.

Given $\delta \in \{(x, y) : \ -R \leq x \leq R, -R \leq y \leq R\}$ and adaptation range R, we penalize distortions δ by a prior probability $p(\delta)$. Thus, Equation (5.17) is rewritten to

$$[w_1^N]_{\text{opt}} = \underset{w_1^N}{\operatorname{argmax}} \left\{ p\left(w_1^N\right) \cdot \prod_{t=1}^{T} \max_{\substack{\delta \in \{(x,y):\\ -R \leq x \leq R, -R \leq y \leq R\}}} \left\{ p(\delta)^{\eta} \cdot p\left(f(X_t,\, l_t + \delta) \,|\, w_1^N\right) \right\} \right\} \tag{5.18}$$

where

$$p(\delta) = \frac{\exp(-\delta^2)}{\sum\limits_{\substack{\delta' \in \{(x,y):\\ -R \leq x \leq R, -R \leq y \leq R\}}} \exp\left(-\delta'^2\right)} \tag{5.19}$$

is renormalized to fulfill the requirements of a probability density function and η is a fixed scale factor.

5.2.2 Adaptation Process

Model-driven tracking adaptation is applied as an iterative process. Each step in this process is limited by the chosen adaptation range R. The adaptation region spanned by the adaptation range R is shown in Figure 5.4 as the green circle. If the chosen adaptation range R is too large, the system might confuse the dominant with the non-dominant hand because a feature extracted in adaptation on the non-dominant hand might lead to better recognition scores. If the chosen adaptation range is too small as shown in Figure 5.4 the system is not able to adapt from the location of a

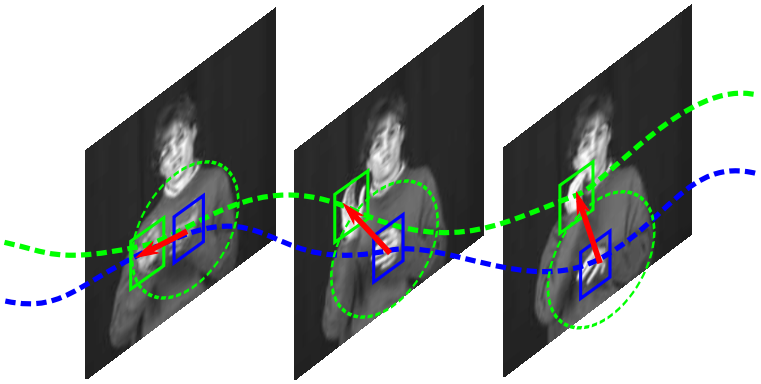


Figure 5.4. Tracking Adaptation On Consecutive Frames

given tracking path shown in blue to the correct location of the oracle tracking path depicted in green. Applied iteratively, model-driven tracking adaptation is able to correct tracking locations as long as parts of the object to be tracked are within the adaptation range as shown by the second image of Figure 5.4. Thus, a given tracking path is fitted to the oracle path w.r.t. to the hypothesized word sequence by applying adaptation iteratively. Nevertheless, once the correct oracle tracking path is outside the adaptation range even iterative tracking adaptation is unable to recover such an error.

The iterative model-driven tracking adaptation process can be used in training to obtain adapted visual models or to bootstrap a model-based tracking framework, and in recognition to obtain adapted tracking paths and features. In both cases, we maximize over the number of iterations K and initialize using the single-best tracking path obtained from the dynamic programming tracking framework described in Section 2.2.

Training

We propose to apply model-driven tracking adaptation iteratively to train visual models. In the two-phase reference system of Chapter 2, the system is trained with features based on the single-best tracking path. If the single-best tracking path contains tracking errors the system learns these tracking errors through the extracted features. Especially appearance-based features are prone to huge changes because of tracking errors. We propose to reduce the impact of tracking errors on the visual models by updating the visual models with additional feature data extracted from the adapted tracking paths.

Given a fixed number of iterations K and the single-best tracking path $l^T_{1,0}$ extracted

from the training data by the dynamic programming tracking framework of Section 2.2, the reference system is trained as follows:

1) INITIALIZE MODELS WITH SINGLE-BEST TRACKING PATH $l^T_{1,0}$

2) ITERATE $k = 1, \ldots, K$ TIMES:
   RECOGNIZE TRAINING DATA AND ADAPT TRACKING PATH $l^T_{1,k-1}$ -> $l^T_{1,k}$
   EXTRACT NEW FEATURE DATA DEPENDING ON $l^T_{1,k}$
   UPDATE MODELS STARTING FROM MODELS OF PREVIOUS ITERATION $k-1$

Recognition

In recognition, we apply model-driven tracking adaptation iteratively. Model-driven tracking adaptation is limited by the adaptation range R. If the object to be tracked is outside the adaptation range R a single adaptation iteration can not recover from such an error. Therefore, we maximize over the number of iterations K to fit the initial single-best tracking path to the trained visual models and the perfect oracle tracking path.

Given a fixed number of iterations K and the single-best tracking path $l^T_{1,0}$ extracted from the test data by the dynamic programming framework of Section 2.2, recognition is performed as follows:

1) INITIALIZE RECOGNITION SYSTEM WITH SINGLE-BEST TRACKING PATH $l^T_{1,0}$

2) ITERATE $k = 1, \ldots, K$ TIMES:
   RECOGNIZE TEST DATA AND ADAPT TRACKING PATH $l^T_{1,k-1}$ -> $l^T_{1,k}$

Chapter 6

Experiments and Results

In this Chapter experimental results for the proposed model enhancement techniques, integrated tracking and recognition, and the proposed approximations to integrated tracking and recognition are presented. Experiments have been performed on the RWTH-BOSTON-104 database described in the following section and are evaluated using the word error rate (WER).

The WER is calculated using the Levenshtein distance [Levenshtein 66] between the true word sequence w_1^N and the recognized word sequence $\hat{w}_1^N$. The distance is defined as the minimum number of edit operations needed to transform one sequence into the other sequence. The WER is invariant whether the true sequence is transformed into the recognized sequence or vice versa. Edit operations are *substitution*, *deletion* and *insertion*.

$$WER = \frac{\#\text{substitutions} + \#\text{insertions} + \#\text{deletions}}{\#\text{reference words}} \cdot 100 \tag{6.1}$$

The Levenshtein distance can be calculated efficiently using dynamic programming [Ney 05].

The proposed integrated tracking and recognition system and the approximations to it have been implemented in the framework of the publicly available[1] RWTH large vocabulary speech recognition system [Lööf & Gollan$^+$ 07].

6.1 RWTH-BOSTON-104 Database

The RWTH-BOSTON-104 database is based on a sign language database for American Sign Language (ASL) published by the National Center for Sign Language and Gesture Resources at Boston University[2]. The database has been recorded with a focus on research on the syntactic structure of ASL [Neidle & Kegl$^+$ 99]. Thus, the data is not optimized for recognition tasks and exhibits a high amount of structural change leading to more realistic data than data recorded especially for sign language recognition tasks. The National Center for Sign Language and Gesture Resources at

[1] http://www-i6.informatik.rwth-aachen.de/web/Software/index.html
[2] http://www.bu.edu/asllrp/ncslgr.html

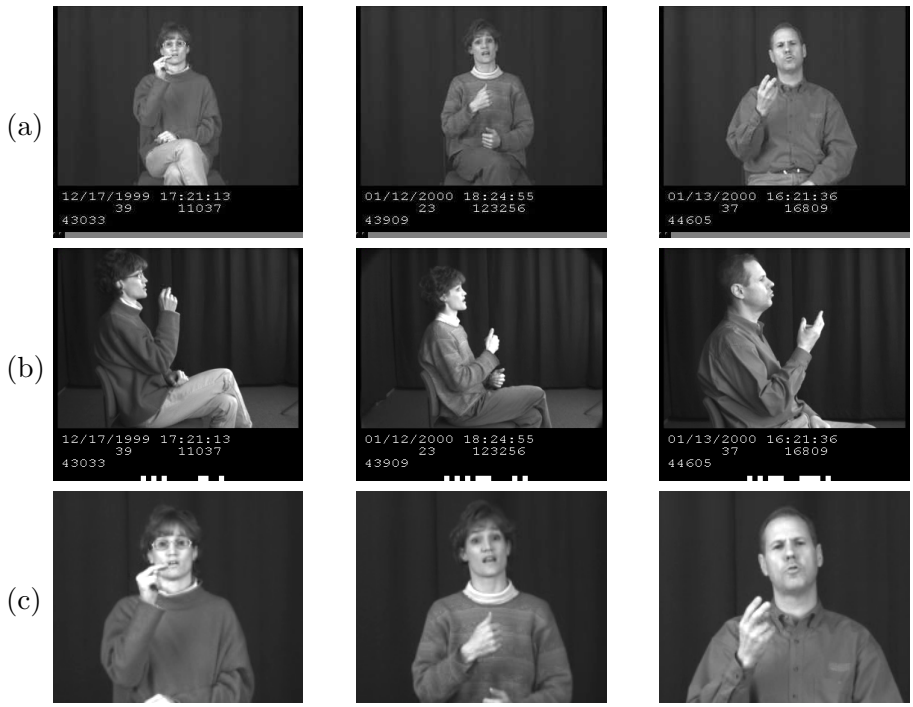


Figure 6.1. Sample frames of the RWTH-Boston-104 database: (a) frontal view, (b) side view, (c) extracted part of the frontal view

Boston University used SignStream^{TM} [Neidle 01, Neidle 02, Neidle 07] to create the original annotations. The Annotations include many aspects of sign language, e.g. pointing gestures.

A set of 201 sentences has been composed for a database by the Human Language and Pattern Recognition Group at RWTH Aachen. The annotation has been revised to fulfill the requirements of a continuous sign language recognition system. The RWTH-BOSTON-104 database is publicly available[3]. The sentences have been performed by three different speakers (two women, one man). The videos have been recorded with four different stationary video cameras. One camera captures the side of the speakers. Two cameras forming a stereo pair show the frontal view of the signers. Moreover, the fourth camera captures only the face of the signers. The videos are recorded at a frame rate of 30 images per second and at a resolution of 312 $\times$ 242 pixels. Sample images are depicted in Figure 6.1.

All experiments presented in the following sections use one of the two frontal cam-

[3]http://www-i6.informatik.rwth-aachen.de/~dreuw/database-rwth-boston-104.php

Table 6.1. Corpus statistics for RWTH-BOSTON-104. OOV means out-of-vocabulary.

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

Table 6.2. Language model statistics for RWTH-BOSTON-104 database.

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

eras. The upper center part (size 195×165) is extracted from each frame because the lower part of the frames contains textual information about the frame and the left and right frame borders are unused.

The database is divided into a training set and a test set consisting of 161 and 40 sentences respectively. All speakers occur in the training and test set. The overall vocabulary of the database comprises 104 words in total with 103 words occurring in the training set. Words only occurring in the test set are called *out of vocabulary* (OOV) words. OOV words cannot be recognized correctly because there is no visual model trained for OOV words. Statistics for the training and test set are presented in Table 6.1. Words occurring only once in the training set are denoted *singletons*. Nine singletons occur also in the test set.

A core problem of the RWTH-BOSTON-104 database is the high number of singletons. Visual models trained with few observations, especially with only one, are unlikely to generalize well for unseen utterances. Moreover, words occurring only once or twice in the training set make up about 45% of all words occurring in the training set (Cf. Figure 6.2). The low number of observations of these words affects the quality of the trained models in general because the estimation of word boundaries depends on the quality of the visual models in continuous sign language recognition.

The perplexity of the test set for different language models estimated on the training set are show in Table 6.2. The logarithm of the perplexity measures the redundancy of the text with respect to the language model. Bigram and trigram perplexities are low because the sentence have a simple structure, e.g. 32 out of 40 sentences in the test set begin with "JOHN".

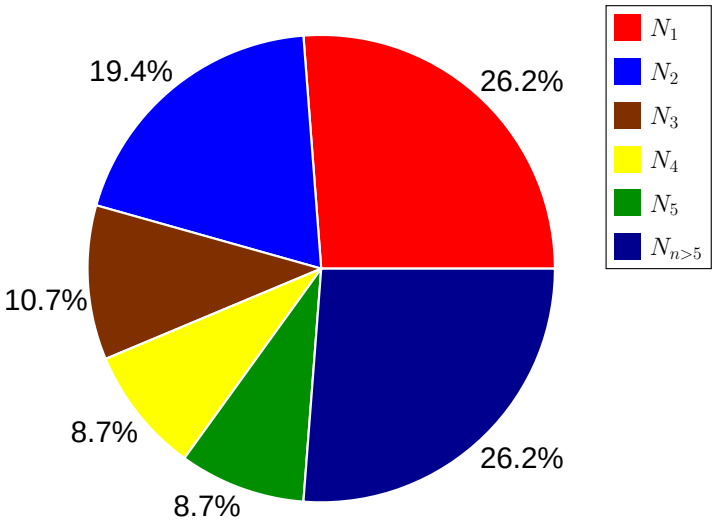


Figure 6.2. Word counts in the training set. $N_n = \{w \mid N_w = n\}$ is the number of words w that occur n-times. N_w is the number of occurrence of word w

6.2 Model Enhancement

We used the model enhancement techniques of Chapter 6.2 to create three additional setups for the RWTH-BOSTON-104 database. The corpus statistics of the RWTH-BOSTON-104 database apply all model enhanced setups too.

In the VSA setup, all training and testing data has been visually aligned. Using VTS with a translation of $\delta \pm 1$ Pixel on the training set of the RWTH-BOSTON-104 database, the VTS setup comprises the equivalent of 9×161 sentences in the training set and 40 sentences in the test set. Hence, each singleton "occurs" at least nine times in the training set of the VTS setup.

Finally, we created the VSA+VTS setup by applying VTS with $\delta \pm 1$ Pixel to the training set of the VSA setup. Hence, the lack-of-data problem in the VSA+VTS setup is reduced as well and each singleton "occurs" nine times in the training set.

In all setups, we use a single state HMM as silence model. All words are modeled by pseudo-phonemes. Each of the 246 pseudo-phonemes is modeled by a three state HMM in Bakis topology as depicted in Figure 2.2 leading to a total of 739 states.

In the following we compare the different setups using appearance-based features. The recognition performance of systems which use statistical models strongly depends on the quality of the trained models because appearance-based features are tested against the models on a pixel level.

Table 6.3 summarizes model statistics for all four setups being trained on 32x32 hand

frame features. In the normal setup, i.e. RWTH-BOSTON-104 database without any enhancement, the system trained 1,185 Gaussian densities for a total of 739 HMM states. 629 states (85.1% of HMM states) are represented by a single density which exceeds the number of singletons (26.2%) in the RWTH-BOSTON-104 database and indicates a poor alignment of observations to states. A problem arises in the normal setup from the high number of Gaussian densities (almost 9% of all trained Gaussian densities) in the silence model. The system performs a nearest-neighbor like decision in the silence model because of the applied Viterbi approximation in recognition. Therefore, the recognition system is dominated by the silence model.

The VSA setup shows a slight increase in the total number of trained Gaussian densities in comparison with the normal setup (1,207 vs. 1,185 Gaussian densities). The absolute number of HMM states represented by single densities is slightly reduced. The increase in the total number of trained Gaussian densities as well as the slight decrease in the number of single densities was expected because VSA reduces the variance in appearance of the different speakers leading to more speaker independent alignments.

The obtained model statistics from the VTS and VSA+VTS setups show a high number of trained Gaussian densities and a low number of states which are represented by a single density. Both findings are to be expected because the amount of data in training has been increased by a factor of nine leading to more observations being assigned to each Gaussian density. The VTS system is still speaker dependent but more robust Gaussian mixture models are estimated due to additional virtual training samples whereas the VSA+VTS setup is also speaker independent. In both VTS setups only 1.9% of the total number of trained Gaussian densities is assigned to the silence model.

Hence, better alignments are generated in the VTS and VSA+VTS setups leading to more robust Gaussian mixture models. Furthermore, singletons are better aligned too leading to more observations being aligned to the word models instead of the silence model.

To visually validate the findings of Table 6.3, we show selected trained Gaussian density means trained from 48x48 frame features in Figure 6.3. Gaussian density means trained from 32x32 hand frames features, and from PCA transformed 40x40 hand frames are visualized in Figure 6.4.

Figure 6.3 shows a selected number of Gaussian density means trained for the pseudo-phoneme JOHN-P0 estimated from 48x48 frame features. In the normal setup (row (a)) and in the VTS setup (row (c)) all three different speakers which are present in the RWTH-BOSTON-104 database are represented by different trained means, i.e. column two represents the male speaker, and columns one and three the two female speakers. Since the recognition system chooses the best fitting density for each feature at each time frame t, recognition systems trained with the normal or VTS setups are

Table 6.3. Model statistics for RWTH-BOSTON104 database trained using 32x32 hand frame features and 246 pseudo-phonemes with 3 states each and a single state silence model; states modeled by mixture densities

	Normal	VSA	VTS	VSA+VTS
#Trained Gaussian densities	1185	1207	6440	6449
#Single density HMM states	629	623	160	139
Fraction of total HMM states	85.1%	84.3%	21.6%	18.9%
#Gaussian densities in silence model	106	118	128	128
Fraction of trained densities	8.9%	9.7%	1.9%	1.9%

speaker dependent. In contrast to the normal and VTS setups, the different speakers are not distinguishable in the trained density means of the VSA and VSA+VTS setups (rows (b) and (d)) clearly indicating speaker independence.

Figure 6.4 shows the first Gaussian density mean of the first state of pseudo-phoneme THROW-P1 being estimated from appearance-based hand frame features. In the normal setup (row (a)), the trained hand model is extremely blurry and the hand is not recognizable as such. Moreover, the hand is estimated to be in the lower right corner of each hand frame feature. Both effects arise from the different scaling and general positioning of the speakers to the camera in the RWTH-BOSTON-104 database. The male speaker is nearer to the camera than the two female speakers leading to a large difference in scale. Throughout the complete RWTH-BOSTON-104 database, the hand of the male speaker is almost of the size of the womens' heads. Therefore, the tracking paths extracted by the dynamic programming framework of Section 2 encode the variance in scale and relative positioning of the hand leading to the depicted blurry Gaussian density mean for 32x32 hand frame features. The blurry model and the shift of the trained hand model to the lower right corner of the Gaussian density mean are also visible in the depicted back projected PCA means because the PCA transformation keeps dimensions with high variance and discards dimensions with low variance in appearance. In the shown PCA means a bright color indicates high variance. Hence, the trained Gaussian densities of the normal setup poorly represent the appearance of the hands and corresponding signs.

In contrast to the normal setup, the hand is clearly recognizable with sharp edges in the depicted Gaussian density means estimated in the VSA (row (b)), VTS (row (c)), VSA+VTS (row (d)) setups. Moreover, the hand is estimated to be centered in the hand frame features for 32x32 hand frames and the PCA transformed 40x40 hand frames. Therefore, VSA and VTS increase the quality of the trained models leading to a better representation of the hands.

All depicted Gaussian density means in rows (b) to (d) feature a dark border artefact

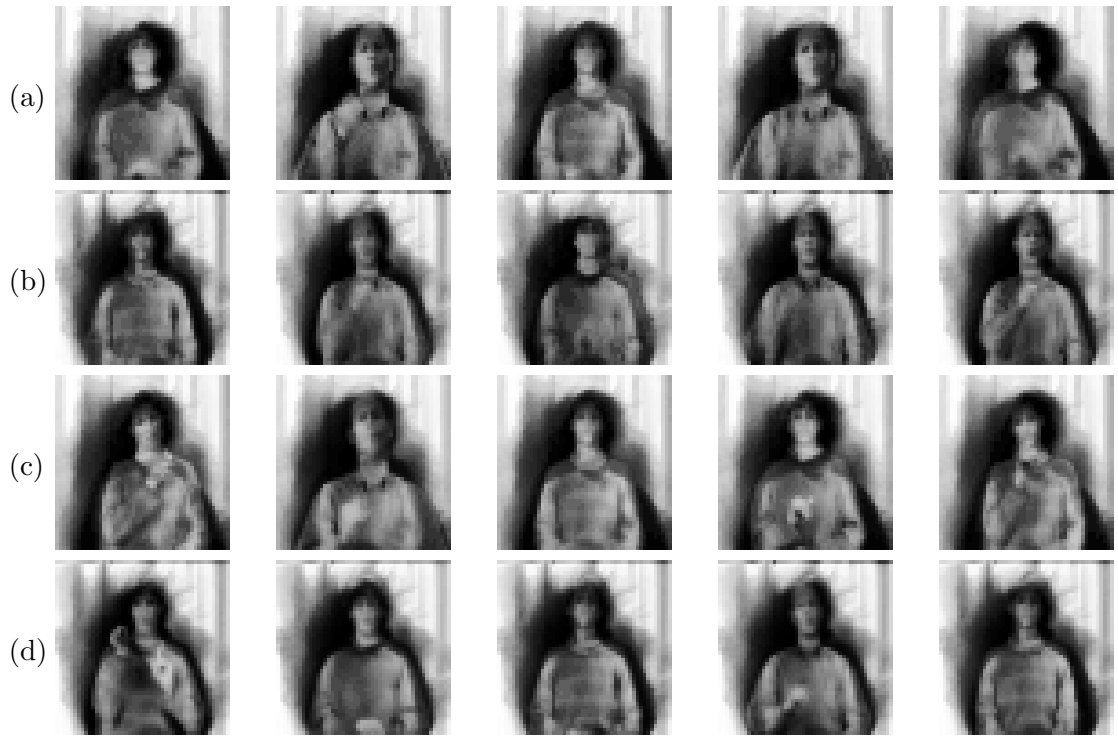


Figure 6.3. Trained density means from 48x48 frame features for state 1 of pseudo-phoneme JOHN-P0 after split 7; intensities histogram normalized (a) standard training, (b) visual speaker alignment, (c) virtual training samples, (d) visual speaker alignment + virtual training samples

in the lower part. If a calculated tracking location at a time frame t is close to the image border the tracking window can include coordinates outside the image which are represented by black pixels. Thus, the system is also trained on observations containing black borders or other tracking related artefacts in all setups. The black artefact border is also present in the normal setup but not in the selected example. In the VSA setup, the local variance of the hand and body is increased because all speakers are centered in the image, and transformed to the same scale and baseline depth. Therefore, also the black border artefacts are aligned and gain a stronger influence on the estimated models.

Comparing the shown Gaussian density mean trained from 32x32 hand frames in the VTS setup to the corresponding density mean in the VSA setup, the hand is not as centered as in the VSA setup and has a larger scale. Reviewing the PCA transformed Gaussian density means of the VSA setup, the variance in appearance is mainly encoded in the hand model and no characteristic background information is

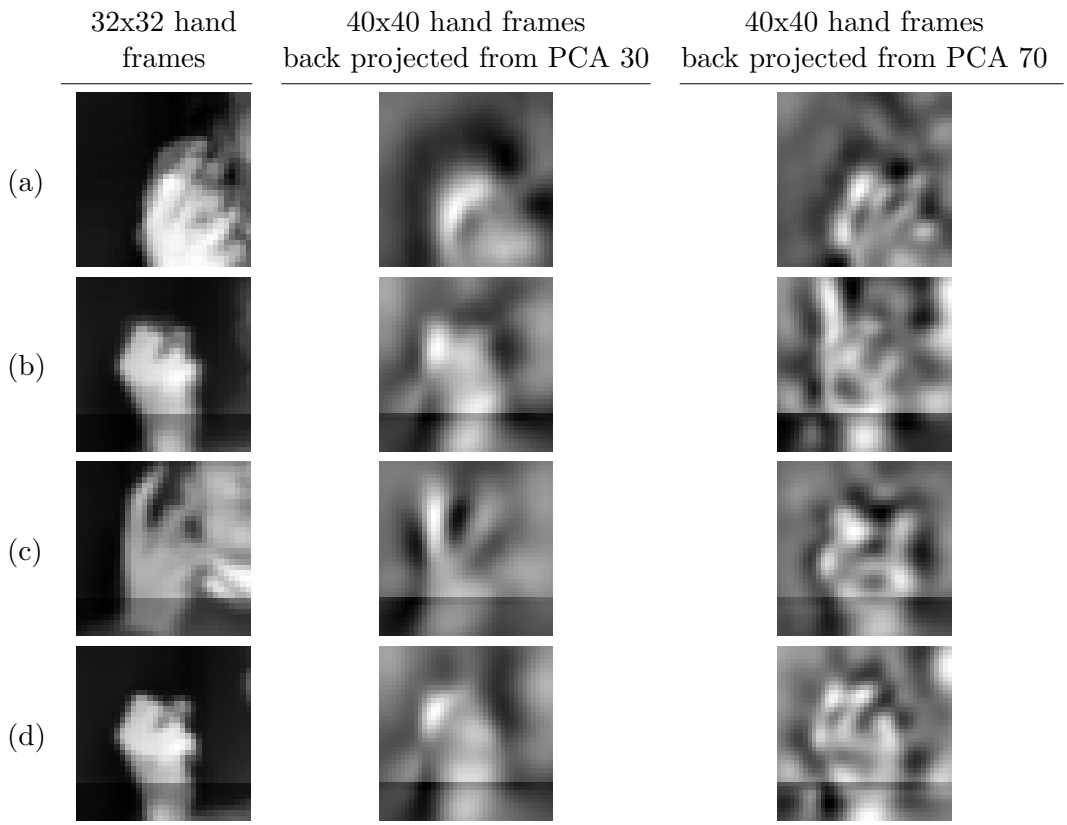


Figure 6.4. Trained density means from hand features (a) standard training, (b) visual speaker alignment, (c) virtual training samples, (d) visual speaker alignment + virtual training samples for pseudo-phoneme THROW-P1 with and without PCA transformation

learned. The PCA transformed density means of the VTS setup show brighter colors, i.e. increased variance in appearance, than the PCA transformed density means of the normal setup in the vicinity of the estimated hand. This has been expected because VTS generation adds global variance to the training dataset by shifting the ROI one pixel in each direction. Due to the added global variance in appearance, the black border artefact is less distinctive in the VTS setup than in the VSA setup.

Finally, the shown Gaussian density mean in the first column of the VSA+VTS setup (row (d)) is visually identical to the depicted density mean of the VSA setup but the black border artefact is less distinct because VTS add variance the appearance of the hands.

Summarizing, Figures 6.3 and 6.4 show that the proposed model enhancement techniques lead to qualitatively better aligned models. VSA leads to trained models which

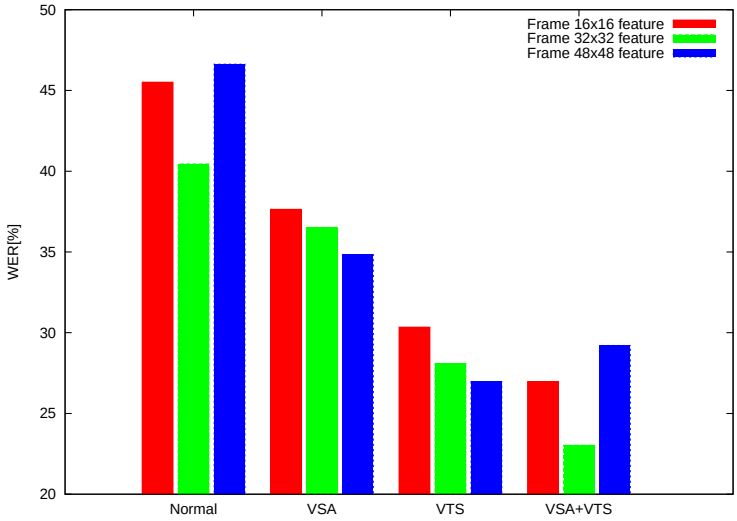


Figure 6.5. Overview of frame feature results for normal, VSA , VTS , and VSA+VTS setups
16x16 results at language model scale 200, word penalty -125, silence penalty 3
32x32 results at language model scale 800, word penalty -500, silence penalty 3
48x48 results at language model scale 1800, word penalty -1350, silence penalty 3

are speaker independent while VTS increases the models' robustness.

In the following, we use appearance-based frame features of different resolutions to quantitatively analyze the effects of the proposed model enhancement techniques to rule out effects induced by tracking errors. To be comparable to previous experimental results presented in [Zahedi 07], we initially choose frame features of resolutions 16x16, 32x32, and 48x48 pixel.

Figure 6.5 shows experimental results for 16x16 frame features being depicted by the red bar, 32x32 frame features being depicted by the green bar, and for 48x48 frame features being depicted by the blue bar. Comparing the three model enhancement setups to the normal setup, the WER is significantly reduced in all experiments using either model enhancement technique. The obtained results confirm that the trained models are significantly improved by applying VSA and VTS. The used reference settings for language model scale, word exit penalty, and silence exit penalty have been empirically optimized on the normal setup. VTS and VSA+VTS setups feature a significantly higher number of trained Gaussian densities than the normal setup. Hence, results of both setups can be due to local maxima/minima in Figure 6.5.

Therefore, we investigate the influence of the language model scale on the recognition performance for 32x32 and 48x48 frame features. Figure 6.6 shows results for all four setups using 32x32 hand frames reduced to 200 dimension after PCA transformation.

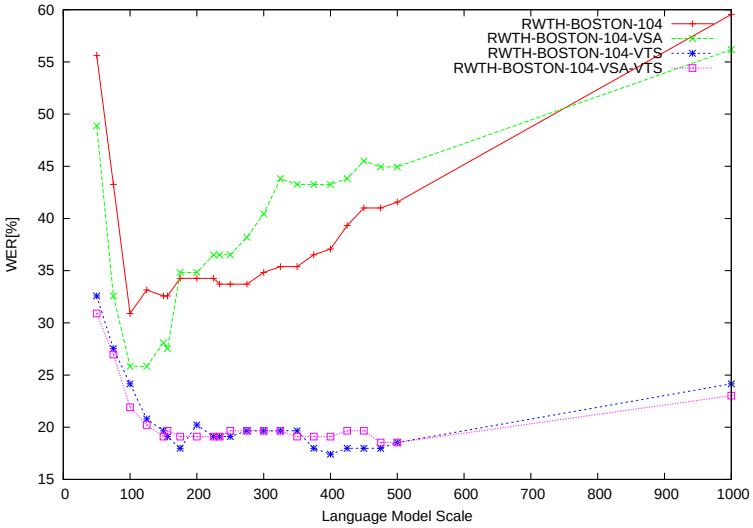


Figure 6.6. Effect of language model scale for 32x32 frame features PCA reduced to 200 dimensions at word penalty -97 and silence exit penalty 3; RWTH-BOSTON-104 vs RWTH-BOSTON-104 + VSA +VTS after 7 split iterations

The word exit penalty of -97 and the silence exit penalty of 3 have been empirically optimized for the normal setup. The results depicted in Figure 6.6 are qualitatively comparable to experiments performed without PCA.

First, we examine the normal setup which is shown as the red line and the VSA setup which is depicted by the green line. The VSA setup outperforms the normal approach up to a language model scale of 175 and is outperformed by the normal setup at language model scales higher than 175. The models trained in the VSA setup are less blurry and lead to lower scores in recognition because of better aligned models. Hence, the influence of the language model is stronger in the VSA setup than in the normal setup at equal language model scale. The language model gains too much influence in the VSA setup at language model scales above 175 introducing additional recognition errors. The VTS and VSA+VTS setups are depicted by the blue and magenta curves respectively in Figure 6.6. Both setups lead to significantly lower WERs than the normal and VSA setup for all shown language model scales because of stronger speaker dependence. There is no significant difference in recognition performance between the blue and magenta curve for the whole spectrum of shown language model scales. Similar results are depicted for 48x48 frame features reduced to 200 dimension using PCA transformation in Figure A.1.

The generation of VTS increases the number of observations in training by a factor of nine. Hence, the total number of trained Gaussian densities in the VTS and the

VSA+VTS is significantly higher than the number of trained Gaussian densities in the normal and VSA setup as shown in Table 6.3 if Gaussian densities are split using the same minimum/maximum observations per density threshold. Therefore, we need to compare the four setups at about the same number of trained Gaussian densities in order to quantitatively evaluate the quality of the trained models.

Figure 6.7 shows the recognition performance of all four setups plotted over the number of splitting iterations in training for 32x32 frame features reduced to 200 dimensions using PCA. Language model scale, word exit penalty, and silence exit penalty are set to system defaults.

The WER of the normal setup (red curve) decreases up to split two and increases for consecutive splits. The increase in WER is due to blurry Gaussian density means because of the lower number of observations per density. The VSA setup (green curve), the VTS setup (blue curve), and the VSA+VTS setup (magenta curve) show a reduction in WER over the shown split iterations because the trained models become more and more speaker dependent in consecutive splits.

The normal and VSA setup comprise about 1,200 trained Gaussian densities after seven split iterations and the VTS and VTS comprise about 1,300 trained Gaussian densities after three split iterations. Therefore, we compare the two non VTS setups at split seven to the VTS setups at split three. The normal setup is clearly outperformed by the three other setups. Furthermore, the VSA setup performs worse than the setups utilizing VTS showing that the VSA is speaker independent in comparison to the VTS setups. The VTS and VSA+VTS setups show similar performance at split three with a slight advantage of the speaker dependent VTS setup.

Until now, we analyzed the effects of model enhancement techniques for appearance-based frame features because appearance-based frame features do not depend on accurate tracking in feature extraction and hence allow an unbiased comparison of the four setups. The focus of this work is the integration of tracking and recognition. Therefore, we use appearance-based hand frame features which depend on accurate tracking in the experiments of Sections 6.3, 6.4, and 6.5. In the remainder of this Section, we analyze the effects of VSA and VTS on appearance-based hand frame features with and without PCA dimension reduction and at different resolutions.

First, we present baseline results for 32x32 hand frame features in Table 6.4 in order to be comparable to previous results presented in [Rybach 06, Dreuw & Rybach$^+$ 07]. Table 6.4 shows that the recognition baseline result of the reference system is improved in the VSA and VSA+VTS setups by more than 10% and in the VTS setup by more than 20%. The huge difference between the VTS and VSA+VTS setups indicates a strong speaker dependence in the VTS setup while the VSA+VTS setup is speaker independent.

Figure 6.8 shows results obtained for PCA transformed 40x40 hand frame features for all four setups. Previous results by [Rybach 06] show that the PCA transformation

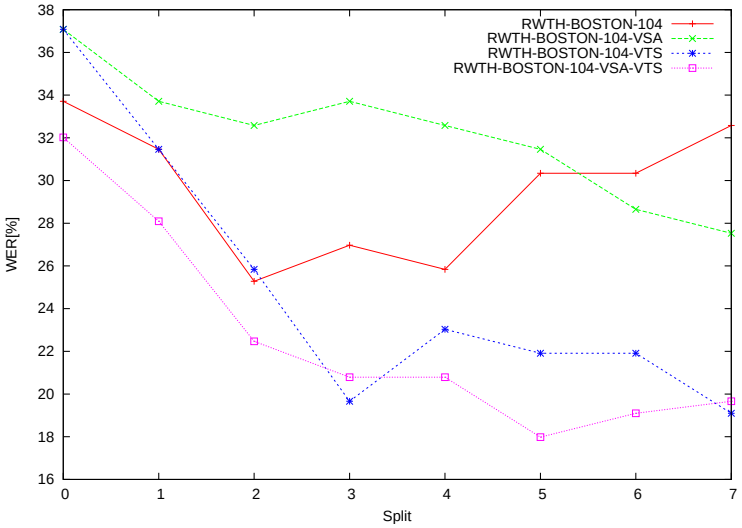


Figure 6.7. Effects of density splitting for 32x32 frame features reduced to 200 dimensions by PCA at language model scale 156, word penalty -97 and silence exit penalty 3

Table 6.4. WER[%] for 32x32 hand features obtained at language model scale 800, word exit penalty -500 and silence exit penalty 3

Normal	VSA	VTS	VSA + VTS
45.51	34.83	**25.28**	31.46

of 40x40 hand frame features outperforms the PCA transformation of 32x32 or 48x48 hand frame features.

The normal setup (red curve in Figure 6.8) shows a global minimum in WER at a reduction to 30 dimensions. At 30 dimensions and lower only the image coordinates with the highest variance in appearance remain. The trained models have been shown to be blurry for the normal setup in Figure 6.4. A reduction to a higher number of dimensions leads to reduced recognition performance in the normal setup because additional distortion i.e. noise is added to the models. The VSA setup performs best at 10 and 20 dimensions and loses recognition performance for higher dimensions. VSA shifts variance in appearance to local points in the training observations and features. Therefore, characteristic differences in appearance are encoded in only a low number of dimensions. In contrast to the normal and VSA setup, the VTS (blue curve) and VSA+VTS (magenta curve) setups perform best at 70 and 90 dimensions respectively. VTS increase the global variance of object appearance and thus adds variance to the trained models of each reference word in both VTS setups. Hence, more dimensions

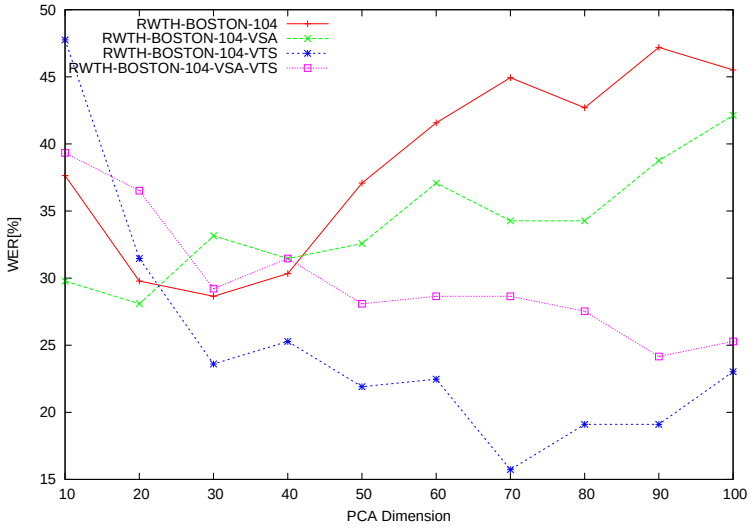


Figure 6.8. Effects of number of PCA reduction from 40x40 hand frames

are required to encode characteristic appearance differences between signs reliably in the VTS and VSA+VTS setup. Starting with 50 dimensions, the normal setup is outperformed by the VSA setup which is outperformed by the VSA+VTS setup. Moreover, the VSA+VTS setup is outperformed by the VTS setup at all depicted PCA dimension values. This effect validates our findings for PCA transformed frame features that applying VSA leads to speaker independent systems while VTS increase the speaker dependence. Therefore, the performance the VSA+VTS setup ranges between the performance of the VSA and VTS setups.

The proposed model enhancement techniques lead to stronger improvements for appearance-based hand frame features than for appearance-based frame features because frame features encode more information than hand frame features. Systems trained on appearance-based frame features learn information regarding the general speaker appearance such as body positioning, head position, clothing and background. In contrast to frame features, hand frame features only encode information regarding the appearance of a hand and characteristic background information. Therefore, models trained on appearance-based hand frame features benefit stronger from the proposed model enhancement techniques.

Summarizing, VSA and VTS improve the recognition performance in continuous sign language recognition. VSA leads to speaker independent systems while VTS leads to more robust Gaussian mixture models. The long time research goal is to create sign language recognition systems which recognize sign language speaker independently.

Therefore, VSA is a promising approach to obtain speaker independent systems.

6.3 Integrated Tracking And Recognition

We tested the proposed integrated tracking and recognition system (Cf. Table 6.5) of Chapter 4 on the VSA+VTS setup of the RWTH-BOSTON-104 database. Language model scale, word exit penalty and silence exit penalty have been optimized empirically using the reference system with normalized scores because the time complexity of the integrated tracking and recognition approach has been shown to be high in Section 4.6. A baseline result of 71.35% WER has been obtained for the reference system of Chapter 2 at the empirically optimized system parameters and an unigram language model. Bigram and trigram results have been obtained through language model rescoring on the word graph generated in the unigram case because the integrated tracking and recognition approach has been implemented only for the unigram case. Using language model rescoring, the baseline is improved to 46.07% and 48.88% in bigram and trigram case respectively at a language model scale of ten.

The integrated tracking and recognition system has a high time complexity. Therefore, strong pruning is required in the integrated tracking and recognition approach introducing additional recognition errors. We opted to apply pruning on the tracking level at a threshold of 0.005 for normalized tracking scoring functions because all word hypotheses are equally affected by pruning on the tracking level in the integrated tracking and recognition approach. Moreover, pruning on the tracking level can be applied to the reference system too because the dynamic programming tracking framework applied in the feature extraction phase allows for pruning. To evaluate how strongly the pruning parameters influence the peformance of the system, we performed similar experiments with the baseline system.

Applying a tracking pruning threshold of 0.005 leads to a WER of 100.56% for the reference system in the unigram case, to 76.40% WER after rescoring the word graph with a bigram language model at language model scale 10, and to a WER of 94.94% after rescoring with a trigram language model. Baseline results with and without pruning are summarized in Table 6.5.

We performed experiments at different tracking model scales γ (Cf. Equation (4.14)) in the integrated tracking and recognition approach to measure the impact of the additional tracking model on recognition. Table 6.5 shows results of the integrated tracking and recognition approach for tracking model scales 1 to 5. The best result of 78.09% (printed in bold face) WER has been achieved at a tracking model scale of three. Comparing the results of the integrated tracking and recognition approach to the result obtained from the reference system at equal pruning settings, the integrated tracking and recognition system clearly outperforms the reference system. The significant improvement of the integrated tracking and recognition system over the reference

Table 6.5. Results on RWTH-BOSTON-104 database in VSA+VTS setup for language model scale 2, word penalty -5, and silence exit penalty 3 utilizing normalized scores and tracking pruning at threshold 0.005; bigram and trigram results obtained from language model rescoring with language model scale 10; Baselines obtained from word conditioned tree search, 32x32 hand features extracted from 177x144 images

Language Model	Baseline without pruning	Baseline with pruning	WER[%] for tracking model scale γ 1	2	3	4	5
unigram	71.35	100.56	85.96	83.71	**78.09**	82.58	79.21
bigram	46.07	76.40	62.36	67.42	64.04	67.42	72.47
trigram	48.88	94.94	73.03	73.60	71.91	70.22	69.66

system at equal pruning settings shows that a continuous sign language recognition systems benefits from an optimization of the tracking path w.r.t. the hypothesized word sequence.

Figure 6.9 shows three examples images from the VSA+VTS setup with visualized tracking decisions (yellow rectangles) reached by the reference system (top row) and the integrated tracking and recognition system (bottom row) at equal tracking pruning settings.

The integrated tracking and recognition system leads to better tracking decisions than the reference system at the given pruning settings. In the top row, the reference system shows a very typical behavior of the hand tracking framework: The hand is not necessarily centered in the single-best tracking path because of the smoothness constraints. In the bottom row, the hand is centered in the tracking window because of model-based adaptation of the tracking path in the integrated tracking and recognition system.

Thus, the integrated tracking and recognition extracts appearance-based features which better resemble the trained models and hence lead to better scores in recognition.

The difference in WER between the baseline result with and without tracking pruning is solely due to tracking errors. Still, the globally best tracking path w.r.t. the tracking criterion leads to the best recognition performance in Table 6.5. The results of the integrated tracking and recognition system indicate that the performance of the reference system can be improved by model-based adaptation of the tracking path because the core difference between both systems is the optimization of the tracking path w.r.t. the hypothesized word sequence in the integrated system. Model-driven adaptation of the tracking path is investigated in Section 6.5.

Although strong pruning on the tracking level is applied, the integrated tracking and recognition system still takes three days of computing time to recognize a single

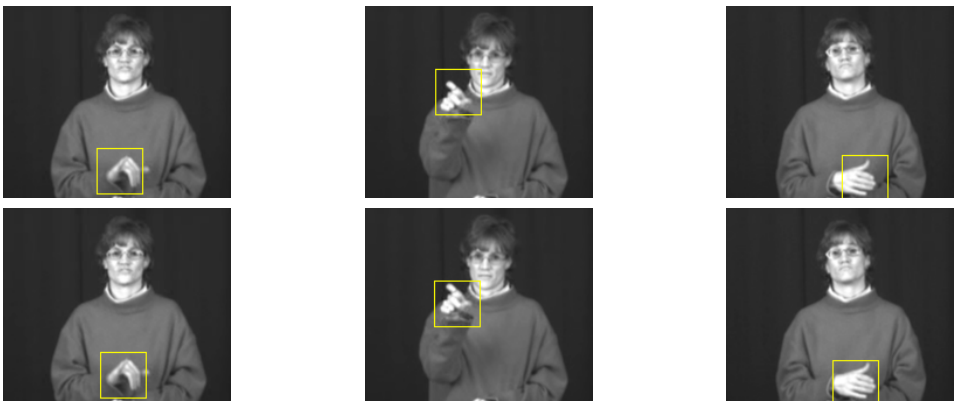


Figure 6.9. Example tracking results of (top row) the reference system and (bottom row) integrated tracking and recognition system at language model scale 2, word exit penalty -5, silence exit 3 and equal tracking settings

sentence of about 100 frames. The reference system needs only a couple of minutes to recognize the same sentence on the same machine including the calculation of the required tracking path for the given sentence. Therefore, efficient approximations to the integrated tracking and recognition system are investigated in the following sections.

6.4 Approximation by Tracking Path Rescoring

In this section, we present several results for rescoring-based approximations to the proposed integrated tracking and recognition system. Experiments have been carried out in the reference system because of the high time complexity of the integrated tracking and recognition system.

The dynamic programming tracking framework of Section 2.2 applies several smoothing parameters, e.g. maximum jump between tracking hypotheses in consecutive frames, ensuring a smooth transition of tracking hypotheses from time t to $t+1$. These smoothness restrictions lead to additional tracking errors for several video sequences in the RWTH-BOSTON-104 database due to abrupt movements of the dominant hand or head. Hence, the globally optimal tracking path w.r.t. the tracking criterion contains errors and does not recover from these errors in most cases because of the smoothness restrictions.

Figure 5.1 shows that the correct locations of the hand are encoded in less optimal tracking paths for the same video sequence. We accept a loss in tracking performance w.r.t. the tracking criterion if we can recover from tracking errors.Assuming a constant background and motion history, i.e. Equation (2.12), in dynamic programming

Table 6.6. Influence of different traceback delays on recognition performance for combination of trajectory and motion features in the reference system at language model scale 80, word exit penalty -45, silence exit 3 and weighting trajectory 0.25 to 0.75 motion and tracking on 195x165 images

	WER[%] for Delay[#Frames]						
	1	5	10	20	60	100	Full
zerogram	51.69	51.69	51.69	52.25	52.25	52.25	52.25
unigram	46.07	46.07	46.07	46.07	46.07	46.07	46.07
bigram	30.90	30.90	31.46	31.46	31.46	31.46	31.46
trigram	**20.22**	20.22	20.22	20.22	20.22	20.22	20.22

tracking, a maximum jump-width of ten pixel and full traceback, we observed that all tracking hypotheses recombine to the single best tracking path after three frames.

Therefore, we ease the smoothness restrictions of the maximum jump-width between consecutive time frames and the full traceback delay. Table 6.6 shows the effect of the eased smoothness restrictions on the recognition performance for a model combination of trajectory and motion features at different language model types. We chose the model combination setup because the best known recognition result of 17.98% WER on the RWTH-BOSTON-104 database is achieved using this setup [Dreuw & Rybach$^+$ 07]. Despite easing the smoothness constraints of the tracking framework, the recognition performance of the model combination setup is constant for all jump-widths and traceback delays. Trajectory and motion features are calculated over a window of five frames, and the feature dimensionality is reduced to 100 dimension by PCA. Therefore, the influence of inaccurate tracking on the recognition performance is strongly reduced in the chosen model combination setup.

The model combination setup achieves equal performance of 20.22% at a maximum jump-width of one pixel and a traceback delay of one frame and at a jump-width of ten pixel and full traceback delay. Therefore, the combination of trajectory and motion features is suitable for a possible real-time continuous sign language system.

Although trajectory and motion features are suitable for real-time applications, the results of Table 6.6 give no information whether or not tracking paths recombine to the single best tracking path. Therefore, we investigated the influence of eased smoothing restrictions on the recognition performance using appearance-based hand frame features. The results of Table 6.7 (baseline set in bold face) show that the recognition performance of the system is reduced if the smoothness restrictions are eased for the single-best tracking path. Furthermore, the reduction in system performance also shows differences in the single-best tracking paths obtained from the dynamic programming tracking framework. Hence, tracking paths do not recombine

Table 6.7. Influence of different traceback delays Δ on recognition for 32x32 hand frame features using trigram language model

Δ	WER[%] for Jump-Width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	80.34	66.85	65.17	58.43	54.49	54.49	52.25	49.44	48.88	**44.94**
50	74.72	63.48	60.67	56.74	58.43	55.06	55.06	53.93	53.93	53.93
25	70.79	61.80	57.87	56.74	55.62	52.81	53.93	55.62	56.18	56.18
10	69.10	65.73	65.17	61.80	63.48	65.73	60.67	60.11	61.24	63.48
1	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01

to a single-best tracking path if the smoothness restrictions of the dynamic programming tracking framework are suitably reduced.

In the following we present experiments, where we use a 400-best tracking path list for each combination of smoothness restrictions depicted in Table 6.7.

6.4.1 m-best Path Rescoring

Using the extracted 400-best tracking path list, we performed rescoring experiments for the normal RWTH-BOSTON-104 setup.

Figure 5.1 shows that the single-best tracking path contains tracking errors. Furthermore, tracking path segments which are erroneous in the single-best tracking path might be correct for tracking paths which are not optimal w.r.t. the tracking criterion. As stated before we accept reduced tracking performance due to eased smoothness constrains as long as the correct tracking path is among the paths of the 400-best tracking path list.

In order to maintain as much tracking context information as possible, we choose to rescore over complete tracking paths rather than single video segments. Table 6.8 shows rescoring results over the top ten and every tenth path from path ten up to path 350. Except of the top ten paths, only every tenth path is used because tracking paths with similar total tracking scores differ only in the first and last frames of the given video sequence.

The baseline result of 44.94% (set in bold face in Table 6.8) is not improved by m-best rescoring. The reference system is only outperformed at severely weakened tracking smoothness restrictions e.g. 48.31% WER vs. 56.18% WER of the reference system at a traceback delay of 25 frames and a maximum jump-width of nine pixel.

The gain in recognition performance at weakened smoothness restrictions indicates

Table 6.8. Results m-best path rescoring over paths 0 to 10 and $10, 20, \ldots, 350$ for lw 800, wp -500, sileex 3, jump penalty 0.3

Δ	WER[%] for jump-width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	76.97	64.61	63.48	57.30	52.25	50.00	52.25	48.31	48.88	**44.94**
50	71.91	61.24	58.43	53.37	54.49	*50.56*	52.81	50.00	51.12	51.12
25	64.61	60.67	56.18	53.37	54.49	49.44	48.31	51.69	*48.31*	50.56
10	67.98	64.04	62.36	60.67	58.99	58.43	56.18	53.93	57.87	60.11
1	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71

that tracking paths which are less optimal w.r.t. the tracking criterion are also less optimal than the single-best tracking path in recognition.

Moreover, m-best rescoring outperforms the reference system results of Table 6.7 which are based on the single-best tracking path: At lower traceback delays ($\leq$ 25 frames) the spatial and temporal context of the tracking paths is reduced. Thus, tracking paths which are not optimal w.r.t. the tracking criterion can only outperform the single-best path at singular frames in recognition.

6.4.2 Multiple Hand Hypotheses

In this section we present experimental results for the multiple hand hypotheses rescoring approach as described in Section 5.1.2.

The results of the m-best path rescoring experiments show that the baseline result is not improved because of the spatial and temporal context which is encoded in the tracking paths. In the multiple hand hypotheses approach the spatial and temporal context of the tracking paths is discarded because the system chooses the best fitting tracking hypotheses from all tracking paths in the 400-best tracking path list at each time frame t. Therefore, smoothness constraints do not limit the performance anymore.

Experiments for the multiple hand hypotheses approach have been conducted for all combinations of smoothness restrictions shown in Table 6.8 using the same 400-best tracking path lists. The obtained results are depicted in Table 6.9.

The baseline result (shown in bold face in Table 6.9) is not improved by the multiple hand hypotheses approach. Moreover, multiple hand hypotheses yields worse results than the m-best rescoring approach at traceback delays greater than 25 frames. Multiple hand hypotheses outperforms the reference system and the m-best path rescoring approach for traceback delays smaller than 25 frames. Tracking paths extracted with a traceback of the best partial path after 25 frames contain a high number of track-

Table 6.9. Results multiple hand hypotheses for lw 800, wp -500, sileex 3, jump penalty 0.3, 400 best paths

Δ	WER[%] for Jump-Width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	74.16	66.85	63.48	58.43	52.25	51.69	52.25	49.44	48.31	**44.94**
50	73.60	58.99	61.24	54.49	58.99	51.12	54.49	53.93	51.69	52.81
25	67.98	62.36	53.93	52.81	57.30	*50.56*	51.69	54.49	51.69	50.56
10	54.49	55.06	56.74	54.55	56.18	54.49	53.93	52.25	52.81	54.49
1	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80

ing errors. The results show that the multiple hand hypotheses approach is able to recover tracking errors by locally selecting the best fitting tracking hypotheses and thus reduces the WER of the recognition system at low temporal and spatial tracking context. Still, the combined results from m-best path rescoring and multiple hand hypotheses show that the recognition system benefits from the temporal and spatial context which is encoded in the tracking paths.

Therefore, we propose to weight the candidate tracking hypotheses in multiple hand hypotheses approach with their corresponding tracking score.

6.4.3 Extended Multiple Hand Hypotheses

In this section, we present experimental results for the extended multiple hand hypotheses approach.

Previous results of the m-best tracking path rescoring and multiple hand hypotheses approaches indicate that the temporal and spatial context of the extracted tracking paths is needed to obtain good recognition results. The temporal and spatial context of candidate tracking hypotheses is expressed by the total score of the corresponding tracking path up to the given tracking hypothesis at the given time frame t. We performed experiments for all combinations of smoothing restriction shown in previous experiments using the tracking sores to weight a tracking hypothesis at a given time t. The weighting scale γ of Equation (5.6) has been empirically optimized to a value of $8 \cdot 10^5$. The scaling factor is of high magnitude because tracking scores are first renormalized in the proposed system and it is then applied to visual scores which are not normalized.

Table 6.10 shows experimental results of the extended multiple hand hypotheses approach. In line with the findings of the m-best tracking path rescoring and multiple hand hypotheses approach, the baseline recognition result of 44.94% is not improved

Table 6.10. Results extended multiple hand hypotheses for lw 800, wp -500, sileex 3, jump penalty 0.3, 400 best paths, weighting scale $\gamma = 8 \cdot 10^5$

Δ	WER[%] for Jump-Width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	74.16	67.98	63.48	58.43	52.25	51.69	52.25	49.44	48.31	**44.94**
50	72.47	60.11	57.30	54.49	58.99	51.12	54.49	52.81	51.12	52.25
25	69.66	58.99	57.30	53.37	53.93	48.31	51.12	55.06	52.25	52.81
10	70.79	66.29	67.98	67.42	60.67	65.17	63.48	63.48	65.17	65.73
1	67.98	67.98	67.98	67.98	67.98	67.98	67.98	67.98	67.98	67.98

in the expended multiple hand hypotheses approach. Moreover, the recognition performance of the extended multiple hand hypotheses approach is worse are equal to the performance of the multiple hand hypotheses approach.

Comparing the results of m-best tracking path rescoring, multiple, and extended multiple hand hypotheses to each other, the single-best tracking path which is extracted by the dynamic programming tracking framework at full smoothness restrictions (maximum jump width 10 and full traceback delay) leads to the best recognition result. The single-best tracking path provides a very good initialization for the recognition system. Therefore, we use the single-best tracking path as starting point for further approximations to the proposed integrated tracking and recognition system.

6.5 Model-driven Tracking Adaptation

In this section, experimental results for model-driven tracking adaptation are presented.

Previous experiments for the integrated tracking and recognition system show that model-driven tracking adaptation may increase the recognition performance of the reference system because the system might be able to recover from tracking errors in the recognition phase. Moreover, experimental results for rescoring based approximations to the integrated tracking and recognition system show that the globally best tracking path w.r.t. the tracking criterion should be chosen as starting point in adaptation.

Thus, we investigate model-driven adaptation of the single-best tracking path for the four model enhancement settings presented in Section 6.2 at three different appearance-based hand frame feature setups.

Table 6.11. WER[%] single iteration tracking adaptation for 32x32 hand features, language model scale 800, word penalty -500, silence exit penalty 3

	Normal	VSA	VTS	VSA+VTS
baseline	45.51	34.83	25.28	31.46
3 Pix adapt.	41.51	26.97	19.66	23.03
5 Pix adapt.	34.83	27.53	19.10	21.35
10 Pix adapt.	44.94	30.90	**17.42**	21.35

6.5.1 Single Iteration Tracking Adaptation

In this section we present experimental results for the first iteration of model-driven tracking adaptation for all four setups described in Section 6.2.

The single-best tracking path has been extracted from 177x144 images using standard smoothness restrictions of a maximum jump-width of ten pixel and full traceback delay. Equation (2.12) is used as tracking criterion. Given these settings, the average distance of the single-best tracking path to the correct hand locations averages at about eight pixel over all sentences of the RWTH-BOSTON-104 database [Rybach 06, Table 7.4]. Therefore, we choose to evaluate model-driven tracking adaptation at adaptation ranges $R = 3$, $R = 5$, and $R = 10$.

Table 6.11 shows experimental results for 32x32 hand frame features. In all setups the baseline is significantly improved by model-driven tracking adaptation. In the normal and VSA setups, the WER increases in the case of adaptation range $R = 10$ in comparison to the results obtained at $R = 5$ because the system chooses features at locations that diverge too far from the single-best tracking path. This effect is called *over-adaptation* in the following.

VTS and VSA+VTS setups show no over-adaptation because the trained models are more robust. Using an adaptation range of $R = 10$, 17.42% WER is achieved in the VTS which improves the best known WER of 17.98% on the RWTH-BOSTON-104 database.

Adaptation results for 40x40 hand frame features reduced to 30 and 70 dimensions are presented in Tables 6.12 and 6.13 respectively. Model-driven tracking adaptation improves the recognition performance of the VSA , VTS , and VSA+VTS in the case of reducing hand frame features to 30 dimensions while the recognition performance is decreased in the normal setup. The loss of recognition performance in the normal setup is due to the poor quality of the trained visual model. Furthermore, the normal and the VTS setup show the effect of too strong adaptation in the case of $R = 10$ indicating that the VSA and VSA+VTS setups are more robust at low feature dimensions.

Table 6.13 shows that model-driven tracking adaptation significantly improves base-

Table 6.12. WER[%] achieved in single iteration adaptation for 40x40 hand frames reduced to 30 dimension by PCA at empirically optimized language model scale 21, word exit penalty 13 and silence exit 3 using tracking on 195 x 165 images

	Normal	VSA	VTS	VSA+VTS
baseline	28.65	33.15	23.60	29.21
+ 3 Pix adapt.	32.02	30.34	21.91	23.60
+ 5 Pix adapt.	30.90	30.34	20.79	20.79
+ 10 Pix adapt.	34.27	28.65	21.35	20.79

Table 6.13. WER[%] achieved in single iteration adaptation for 40x40 hand frames reduced to 70 dimension by PCA at empirically optimized language model scale 50, word exit penalty 31, and silence exit 3 using tracking on 195 x 165 images

Feature	Normal	VSA	VTS	VSA+VTS
baseline	44.94	34.27	15.73	26.97
+ 3 Pix adapt.	32.58	29.78	14.61	12.92
+ 5 Pix adapt.	34.83	27.53	12.92	13.48
+ 10 Pix adapt.	56.74	34.83	14.61	**12.92**

line results in all setups at a higher number of remaining feature dimensions. In contrast to the results obtained in the cases of 32x32 hand frame features, and 40x40 hand frame features reduced to 30 dimensions by PCA, results for 40x40 hand frame features reduced to 70 dimensions show over-adaptation at all setups. We achieve a recognition error rate of 12.92% in the VSA+VTS setup which improves the previously observed best error rate of 17.42% by 3.5%.

The single-best tracking path strongly differs from the perfect oracle path if the video sequence contains abrupt movements of the object to be tracked because of the smoothness restrictions in the dynamic programming framework. Setting the problem of over-adaptation aside, model-driven tracking adaptation is expected to recover from tracking errors due to abrupt movements using an adaptation range of ten or even higher. Despite the effect of over-adaptation, we observe improvements with adaptation range $R = 10$ for all features types and at all database setups which we investigated in Tables 6.11, 6.12, and 6.13. We expect improvements over the already presented results if distortions (Cf. Section 5.2.1) are penalized according to their distance to the single-best tracking path.

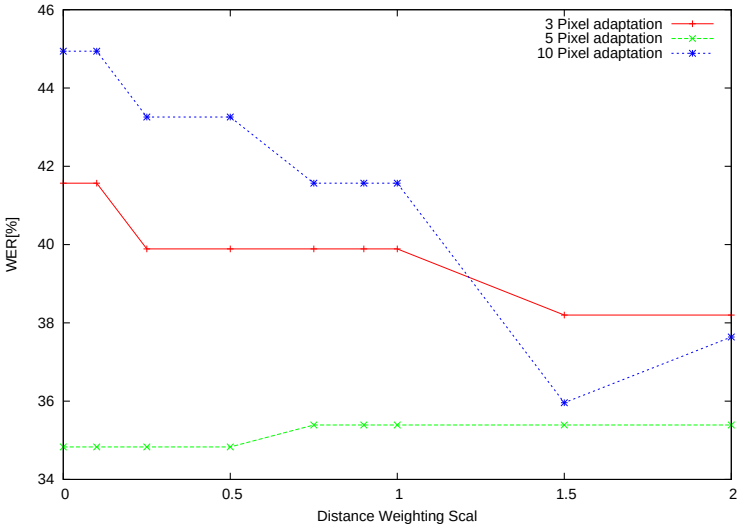


Figure 6.10. Effect of distance weighting scale parameter η on recognition of 32x32 hand frame features at language model scale 800, word exit penalty -500 and silence exit penalty 3 in normal RWTH-BOSTON-104 setup

6.5.2 Distance Weighting

In the following, experimental results for distance weighting in model-driven tracking adaptation are presented. Due to the reasons discussed in the previous section, we expect distance weighting to improve the results from model-driven tracking adaptation without distance weighting.

The influence of distance weighting on model-driven tracking adaptation is controlled by the distance weighting scale parameter η in Equation (5.18). We first empirically optimize the scale parameter η as depicted in Figure 6.10 for the normal setup with 32x32 hand frame features.

The baseline of the investigated adaptation ranges is depicted at scaling parameter $\eta = 0$. Figure 6.10 shows that distance weighting reduces the WER of adaptation ranges $R = 3$ (red curve) and $R = 10$ (blue curve) for all depicted values of the scaling parameter η. Adaptation range $R = 10$ strongly benefits from higher values of η and reaches a minimum at $\eta = 1.5$. The result for adaptation range $R = 10$ validates our previous expectations. The green curve (adaptation range $R = 5$) is almost constant for all depicted values of η and shows that a high influence of distance weighting on model-driven tracking adaptation is desirable.

Table 6.14 shows experimental results obtained for 32x32 hand frame features using distance weighting with $\eta = 1.5$. The upper half the table contains baseline results. Distance weighting either significantly improves system performance over the corre-

Table 6.14. WER[%] single iteration tracking adaptation for 32x32 hand features, language model scale 800, word penalty -500, silence exit penalty 3, distance weighting scale factor empirically optimized to $\eta = 1.5$ for adaptation range $R = 10$ in the normal setup

	Normal	VSA	VTS	VSA+VTS
baseline	45.51	34.83	25.28	31.46
+ 3 Pix adapt.	41.57	26.97	19.66	23.03
+ 5 Pix adapt.	34.83	27.53	19.10	21.35
+ 10 Pix adapt.	44.94	30.53	17.42	21.35
+ 3 Pix adapt. + pen.	38.20	26.97	19.66	22.47
+ 5 Pix adapt. + pen.	35.39	25.84	18.54	21.35
+ 10 Pix adapt. + pen.	35.96	28.65	**18.54**	20.79

sponding baseline or keeps approximately equal performance. Furthermore, the result for adaptation range $R = 10$ is similar to the results obtained for $R = 5$ showing a reduction in over-adaptation. The increase from 25.84% at $R = 5$ to 28.65% at $R = 10$ is due to over-adaptation in the VSA setup.

Experimental results for 40x40 hand frame features reduced to 30 and 70 dimensions by PCA are presented in Tables 6.15 and 6.16 respectively. The distance weighting scale parameter η is optimized to $\eta = 0.25$ in the VTS setup. Scale parameter optimization has been performed in the VTS setup because the VTS setup showed to be the most robust setup in previous experiments on model-driven tracking path adaptation for 40x40 hand frame features with reduced dimensionality.

Over-adaptation is removed by distance weighting for all setups in Table 6.15. Distance weighting significantly improves recognition performance for 40x40 hand frame features reduced to 30 dimensions by PCA at adaptation range $R = 10$ for all setups of the RWTH-BOSTON-104 database. In the VSA , VTS , and VSA+VTS setups the WER is improved by more than 2% at adaptation range $R = 10$ and in the normal setup by more than 4% indicating that the system without distance weighting chooses distortions far away from the single-best tracking path. Hence, the single-best tracking path represents a good initialization of the recognition system.

Experimental results for 40x40 hand frame features reduced to 70 dimensions show similar improvements by distance weighting as results obtained for 32x32 hand frame features and 40x40 hand frame features reduced to 30 dimensions. In Table 6.16, distance-weighting reduces over-adaptation in the VSA and VSA+VTS setup showing that both setups lead to more robust models than the normal and VTS setup respectively. Moreover, the recognition system achieves a WER of 11.24% at adaptation range $R = 10$ in the VTS setups again improving the previously best WER of 12.92%

Table 6.15. WER[%] achieved in single iteration adaptation using distance weighting for 40x40 hand frames reduced to 30 dimension by PCA at at language model scale 21, word exit penalty 13, silence exit 3, and distance weighting scale $\eta = 0.25$ empirically optimized for adaptation range $R = 10$ in the VTS setup using tracking on 195 x 165 images

	Normal	VSA	VTS	VSA+VTS
baseline	28.65	33.15	23.60	29.21
+ 3 Pix adapt.	32.02	30.34	21.91	23.60
+ 5 Pix adapt.	30.90	30.34	20.79	20.79
+ 10 Pix adapt.	34.27	28.65	21.35	20.79
+ 3 Pix adapt. + pen.	32.02	29.78	21.35	23.03
+ 5 Pix adapt. + pen.	31.46	26.97	21.35	20.22
+ 10 Pix adapt. + pen.	30.90	26.97	**17.98**	18.54

Table 6.16. WER[%] achieved in single iteration adaptation using distance weighting for 40x40 hand frames reduced to 70 dimension by PCA at language model scale 50, word exit penalty 31, silence exit 3, and distance penalty scale $\eta = 0.25$ empirically optimized for adaptation range $R = 10$ in the VTS setup using tracking on 195 x 165 images

	Normal	VSA	VTS	VSA+VTS
baseline	44.94	34.27	15.73	26.97
+ 3 Pix adapt.	32.58	29.78	14.61	12.92
+ 5 Pix adapt.	34.83	27.53	12.92	13.48
+ 10 Pix adapt.	56.74	34.83	14.61	12.92
+ 3 Pix adapt. + pen.	33.71	31.46	15.17	13.48
+ 5 Pix adapt. + pen.	30.90	24.72	13.48	14.04
+ 10 Pix adapt. + pen.	32.58	24.16	**11.24**	12.92

by 1.5%.

Summarizing, the recognition performance of the reference system is strongly improved using model-driven tracking adaptation and distance weighting. All conducted experiments in model-driven tracking adaptation with distance weighting reach best results at an adaptation of $R = 10$. Due to abrupt movements of the dominant hand in the RWTH-BOSTON-104 database and the maximum jump-width of 10 pixel, the single-best tracking path can differ by more than 10 pixel from the perfect oracle path. Therefore, we expect further improvements in recognition performance if model-driven

tracking adaptation with distance weighting is applied as an iterative procedure. Starting from the single-best tracking path, the resulting tracking path from the previous adaptation iteration initializes the following iteration of model-driven tracking adaptation with distance weighting. In this way, the system emulates jump-widths of more than 10 pixel without increasing the time complexity of the system.

6.5.3 Multiple Iteration Recognition

In this section, preliminary result for multiple iterations of model-driven tracking adaptation in recognition are presented. Ten iterations of model-driven tracking adaptation are conducted for 32x32 hand frame features with adaptation range $R = 10$ and distance weighing scale factor $\eta = 1.5$ in the normal setup (red curve in Figure 6.11). All system parameters are kept constant at all iterations in order to reliable evaluate the effect of iterative model-driven tracking adaptation. Moreover, the same experiment is conducted for the best known recognition setup so far. The green curve in Figure 6.11 depicts the performance of the recognition system for 40x40 hand frame features reduced to 70 dimensions by PCA using model-driven tracking adaptation at an adaptation range of $R = 10$ and distance weighting at $\eta = 0.25$. In Figure 6.11, the result depicted at iteration number 0 represents the WER obtained in the previously presented single iteration experiments.

The red curves stays constant for one additional iteration of model-driven tracking adaptation and than increases by 2% before going into saturation. The green curve strongly increases from 11% to 55% WER even after one additional iteration of model-driven tracking adaptation with distance weighting and goes than into saturation at over 55% WER. Figure 6.11 clearly shows that model-driven tracking adaptation with distance weighting achieves a global minimum at a single iteration. The strong decrease in recognition performance indicates that the system assumes object locations far away from the initial single-best path to be optimal for the hypothesized word sequence. Hence, the system should only consider distortions in a vicinity of ten pixel around the single-best tracking path obtained from the dynamic programming tracking framework (Cf. Section 2.2).

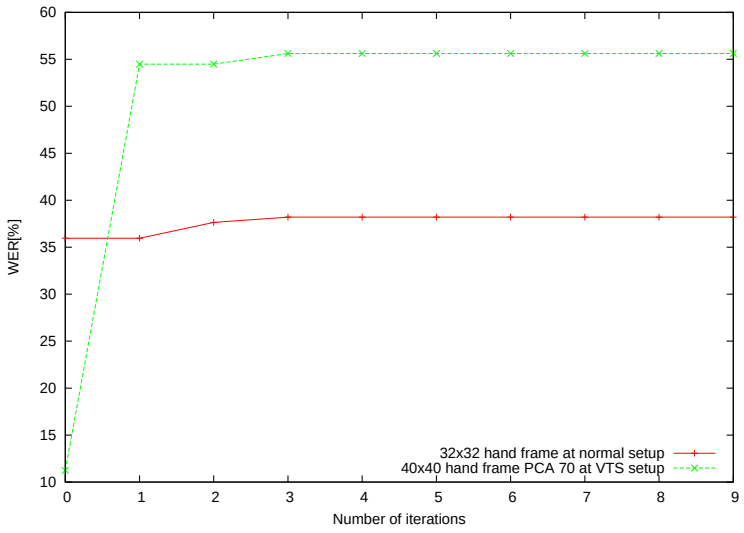


Figure 6.11. Effect of iterative model-driven tracking adaptation in recognition with constant distance weighting scale for 32x32 hand frame features at language model scale 800, word exit penalty -500, silence exit penalty 3, and $\eta = 1.5$ in the normal setup; and 40x40 hand frame features reduced to 70 dimension at language model scale 50, word exit penalty 31, silence exit penalty 3, and $\eta = 0.25$ in the VTS setup

Chapter 7

Conclusion & Perspectives

Conclusion

In the course of this work, we have investigated visual speaker alignment and virtual training samples to overcome the lack-of-data problem in continuous sign language recognition. Both techniques have been shown to increase the robustness of the trained models leading to improved recognition performance. A combination of virtual training samples and virtual training samples lead to a robust and speaker independent recognition system.

An integrated tracking and recognition system for continuous sign language recognition has been implemented and experimentally evaluated. Evaluation of the system has been focussed on appearance-based features because these depend strongly on accurate tracking. Due to the high time complexity of the integrated system strong pruning was necessary. The integrated tracking and recognition system has not reached baseline performance but has been shown to significantly outperform the reference system if equal pruning settings are applied. Experimental results indicate that the recognition performance of the reference recognition system can be enhanced by model-driven tracking i.e. feature adaptation in the recognition phase.

Efficient approximations to the integrated tracking and recognition system have been integrated into the reference system and have been experimentally evaluated. First, m-best tracking path rescoring, multiple hand hypotheses, and extended multiple hand hypotheses approaches have been investigated according to the concept of "acoustic" rescoring known in speech recognition. Although neither method improved the performance of the reference recognition system, experiments showed that single-best tracking paths obtained through dynamic programming tracking should include as much spatial and temporal context as possible. Experimental results for the above approximations showed that the single-best tracking path represents a very good initialization of the recognition system.

Model-driven adaptation of the single-best tracking path has been investigated as approximation to the integrated tracking and recognition system. Experimental results for three different adaptation ranges and three different appearance-based feature setups show significant improvement over state-of-the-art recognition baselines. Combining model-driven tracking adaptation with both model enhancement techniques,

the currently best known WER of 17.98% has been reduced to 12.92% WER using appearance-based hand frame features only. Furthermore, the recognition performance could be further increased to the best currently known WER of 11.24 for the RWTH-BOSTON-104 database by penalizing large distortions in model-driven tracking adaption. Preliminary results for iterative model-driven tracking adaptation show no further improvement because the system diverges too far from the single-best tracking path.

The results obtained in this work show that approximations to a joint optimization of tracking and recognition w.r.t. the hypothesized word sequence significantly improves recognition performance, but that a fully integrated tracking and recognition approach is not feasible due to high time complexity.

Perspectives

Further investigations into the proposed integrated tracking and recognition system are not expected because the approximation of model-driven tracking adaptation leads to very good recognition results.

In this work, model-driven tracking adaptation has been applied in recognition. More robust visual models could be trained using model-driven tracking adaptation in training. Moreover, model-driven tracking adaptation can be used to bootstrap a model-based tracking framework providing more accurate tracking paths especially in noisy environments. A noisy and challenging database for German sign language recognition has been presented by [Bungeroth & Stein$^+$ 06]. To our knowledge no meaningful recognition results exist for this database yet.

The proposed model-driven tracking adaptation approach can be enhanced in several different aspects. Model-driven tracking adaptation should allow for variable adaptation ranges R to cope with strong abrupt movements of the objects, and objects which dynamically enter and leave scenes. Currently the adaptation space of model-driven tracking adaptation is modeled by a rectangle of equal size for all trained reference words. If the position of an object is included in the visual models, the adaptation space can be warped depending on the hypothesized word and the initial candidate location of the single-best tracking path. Further improvements are expected if model-driven tracking adaptation is enhanced by a distinct adaptation model accounting for additional changes in object appearance such as illumination and shape.

Appendix A

Additional Result Tables

Table A.1. Complete Results m-best path rescoring over paths 0 to 10 and $10, 20, \ldots, 350$ for lw 800, wp -500, sileex 3, jump penalty 0.3

Δ	WER[%] for jump-width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	76.97	64.61	63.48	57.30	52.25	50.00	52.25	48.31	48.88	44.94
120	76.97	64.04	62.92	56.18	52.81	50.56	51.69	48.88	48.31	44.94
100	75.28	66.29	64.04	58.43	52.81	51.12	51.69	48.88	47.19	44.94
50	71.91	61.24	58.43	53.37	54.49	50.56	52.81	50.00	51.12	51.12
25	64.61	60.67	56.18	53.37	54.49	49.44	**48.31**	51.69	**48.31**	50.56
10	67.98	64.04	62.36	60.67	58.99	58.43	56.18	53.93	57.87	60.11
5	71.91	71.35	70.79	70.79	70.22	67.98	66.85	67.42	67.98	67.98
4	75.84	75.84	74.16	73.60	74.16	70.79	70.79	69.66	70.22	70.22
3	75.28	73.60	73.03	72.47	74.72	73.60	72.47	72.47	73.60	73.60
2	77.53	74.16	75.84	75.28	75.84	75.84	74.16	74.72	73.60	73.60
1	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71	83.71

Table A.2. Results path 0 for lw 800, wp -500, sileex 3, jump penalty 0.3

Δ	WER[%] for jump-width in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	80.34	66.85	65.17	58.43	54.49	54.49	52.25	49.44	48.88	44.94
120	80.34	65.73	65.17	59.55	55.06	55.06	52.25	51.12	50.00	46.07
100	79.78	68.54	66.29	59.55	55.06	52.25	51.69	50.00	47.75	45.51
50	74.72	63.48	60.67	56.74	58.43	55.06	55.06	53.93	53.93	53.93
25	70.79	61.80	57.87	56.74	55.62	52.81	53.93	55.62	56.18	56.18
10	69.10	65.73	65.17	61.80	63.48	65.73	60.67	60.11	61.24	63.48
5	73.60	71.35	71.91	73.03	72.47	70.79	69.66	71.91	72.47	72.47
4	78.65	75.84	84.27	80.90	74.16	72.47	70.79	69.66	71.35	71.91
3	78.09	73.60	74.16	72.47	75.84	75.28	75.28	73.60	73.60	73.60
2	79.21	79.78	78.65	79.21	79.21	79.21	79.21	79.21	79.21	79.21
1	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01	91.01

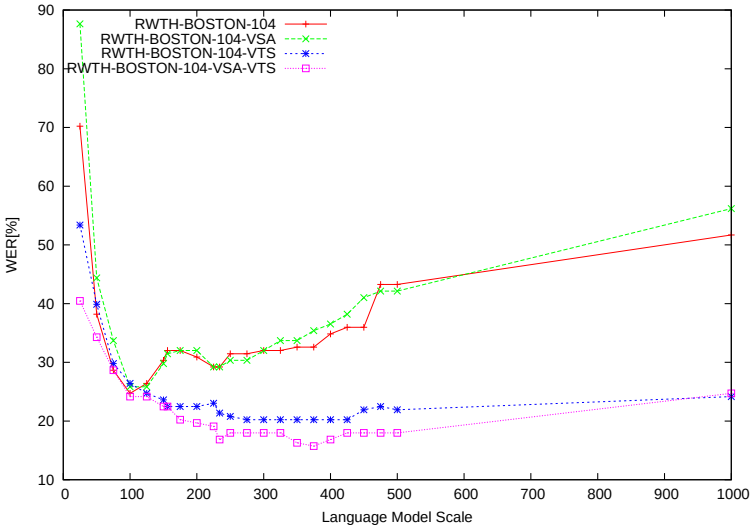


Figure A.1. Effect of language model scale for 48x48 frame features PCA reduced to 200 dimensions at word penalty -97 and silence exit penalty 3; RWH-BOSTON-104 vs RWTH-BOSTON-104 + VSA +VTS

Table A.3. Results multiple hand hypotheses for lw 800, wp -500, sileex 3, jump penalty 0.3, 400 best paths

Δ	WER[%] for JumpWidth in Pixel									
	1	2	3	4	5	6	7	8	9	10
FULL	74.16	66.85	63.48	58.43	52.25	51.69	52.25	49.44	48.31	44.94
100	70.22	67.98	64.61	59.55	55.06	53.37	52.81	50.00	47.75	45.51
50	73.60	58.99	61.24	54.49	58.99	51.12	54.49	53.93	51.69	52.81
25	67.98	62.36	53.93	52.81	57.30	50.56	51.69	54.49	51.69	50.56
10	54.49	55.06	56.74	54.55	56.18	54.49	53.93	52.25	52.81	54.49
5	64.04	67.98	67.42	67.98	65.73	68.54	65.73	67.42	66.85	65.73
4	56.74	58.99	58.99	58.43	60.11	54.55	60.11	60.67	60.67	61.80
3	60.11	65.73	65.17	65.17	64.04	63.48	63.48	62.36	63.48	64.61
2	61.24	61.80	61.24	61.24	61.24	61.24	61.24	61.24	61.24	61.24
1	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80	61.80

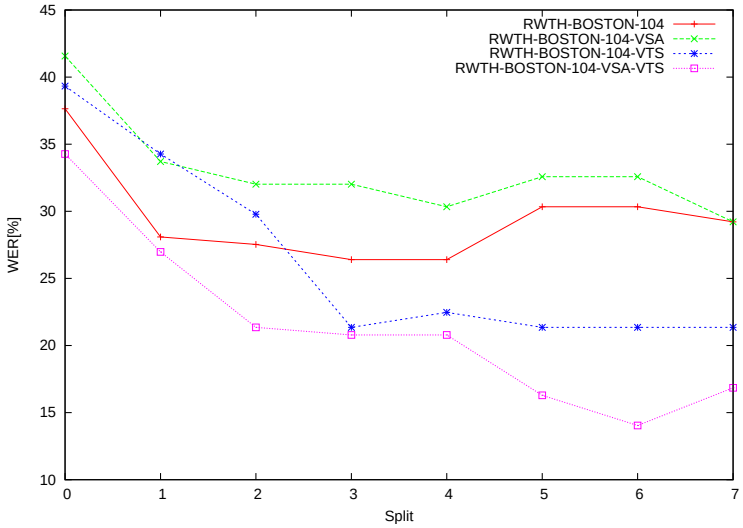


Figure A.2. Effects of density splitting for 48x48 frame features reduced to 200 dimensions by PCA at language model scale 234, word penalty -97 and silence exit penalty 3

Bibliography

[Alon & Athitsos$^+$ 05] J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff: Simultaneous Localization and Recognition of Dynamic Hand Gestures. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, 2005.

[Andriluka & Roth$^+$ 08] M. Andriluka, S. Roth, B. Schiele: People-Tracking-by-Detection and People-Detection-by-Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.

[Arulampalam & Maskell$^+$ 02] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp: A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 174–188, Feb. 2002.

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

[Avidan 07] S. Avidan: Ensemble trracking. In *IEEE Transactions on pattern Analysis and Machine Intelligence*, Vol. 29, pp. 261 – 271, February 2007.

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

[Bauer] B. Bauer: *Erkennung kontinuierlicher Gebärdensprache mit Untereinheiten-Modellen*. Ph.D. thesis, RWTH Aachen University.

[Bauer & Kraiss 01] B. Bauer, K.F. Kraiss: Towards an Automatic Sign Language Recognition System Using Subunits. In *Gesture and Sign Language in Human-Computer Interaction. International Gesture Workshop GW 2001*, Vol. 2298 of *Lecture Notes in Artificial Intelligence*, pp. 64–75, London, April 2001. Springer Verlag.

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

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

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

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

[Berrar & Dubitzky$^+$ 03] D.P. Berrar, W. Dubitzky, M. Granzow: *A Practical Approach to Microarray Data Analysis*. Springer, 2003.

[Bowden & Windridge$^+$ 04] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, M. Brady: A Linguistic Feature Vector for the Visual Interpretation of Sign Language. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, May 2004.

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

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

[Bungeroth & Stein$^+$ 06] J. Bungeroth, D. Stein, P. Dreuw, M. Zahedi, H. Ney: A German Sign Language Corpus of the Domain Weather Report. In *International Conference on Language Resources and Evaluation*, pp. 2000–2003, Genoa, Italy, May 2006.

[Burges & Schöllkopf 97] C.J.C. Burges, B. Schöllkopf: Improving the Accuracy and Speed of Support Vector Machines. In *Neural Information Processing Systems (NIPS 97)*, Vol. 9, pp. 375–385, Vancouver, Canada, dec 1997.

[Chu & Huang 07] S.M. Chu, T.S. Huang: Audio-Visual Speech Fusion Using Coupled Hidden Markov Models. In *CVPR*. IEEE Computer Society, June 2007.

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

[Cooper & Bowden 07a] H. Cooper, R. Bowden: Large Lexicon Detection of Sign Language. *LNCS*, Vol. 4796, pp. 88 – 97, 2007. IEEE Workshop Human Computer Interaction ICCV2007.

[Cooper & Bowden 07b] H. Cooper, R. Bowden: Sign Language Recognition Using Boosted Volumetric Features. In *Proceedings IAPR Conference on Machine Vision Applications*, pp. 359 – 362, May 2007.

[Deselaers & Dreuw$^+$ 08] T. Deselaers, P. Dreuw, H. Ney: Pan, Zoom, Scan – Time-coherent, Trained Automatic Video Cropping. In *IEEE Conference on Computer Vision and Pattern Recognition*, accepted for publication, Anchorage, AK, USA, June 2008. IEEE.

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

[Dreuw & Forster$^+$ 08] P. Dreuw, J. Forster, T. Deselaers, H. Ney: Efficient Approximations to Model-Based Jointed Tracking And Recognition For Continous Sign Language. submitted to IEEE 8th International Conference on Automatic Face And Gesture Recognition, 17. – 19. September 2008.

[Dreuw & Ney 08] P. Dreuw, H. Ney: Visual Modeling and Feature Adaptation in Sign Language Recognition. accepted for publication at the 8th ITG Conference on Speech Communication, 8 – 10 October 2008.

[Dreuw & Rybach$^+$ 07] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, H. Ney: Speech Recognition Techniques for a Sign Language Recognition System. In *Interspeech*, pp. 2513–2516, Antwerp, Belgium, Aug. 2007. ISCA best student paper award Interspeech 2007.

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

[Elgammal 05] A. Elgammal: Learning to Track: Conceptual Manifold Map for Closed-form Tracking. In *Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, June 2005.

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

[Farhadi & Forsyth+ 07] A. Farhadi, D. Forsyth, R. White: Transfer Learning in Sign Language. In *CVPR*, 2007.

[Felzenszwalb & Huttenlocher 05] P.F. Felzenszwalb, D.P. Huttenlocher: Pictoral Structures for Object Recognition. *International Journal of Computer Vision*, Vol. 61, No. 1, pp. 55–79, 2005.

[Gavrila & Philomin 99] D.M. Gavrila, V. Philomin: Real-Time Object Detection For "Smart" Vehicles. In *IEEE International Conference On Computer Vision*, pp. 87–99, Kerkyra, 1999.

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

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

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

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

[Kadir & Bowden+ 04] T. Kadir, R. Bowden, E. Ong, A. Zisserman: Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition. In *Proceedings of he British Machine Vision Conference*, Vol. 2, pp. 939 – 948, 2004.

[Kobayashi & Haruyama 97] T. Kobayashi, S. Haruyama: Partly-Hidden Markov Model And Its Application To Gesture Recognition. *IEEE Internation Conference on Acoustics, Speech, and Signal Processing (ICASSP' 97)*, Vol. 4, pp. 3081–3084, 1997.

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

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

[Lööf & Gollan+ 07] J. Lööf, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter, H. Ney: The RWTH 2007 TC-STAR Evaluation System for European English and Spanish. In *Interspeech*, pp. 2145–2148, Antwerp, Belgium, Aug. 2007.

[Mikolajczyk & Schmid+ 04] K. Mikolajczyk, C. Schmid, A. Zisserman: Human detection based on a probabilistic assembly of robust part detectors. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. sp, may 2004.

[Navaratnam & Thayananthan+ 05] R. Navaratnam, A. Thayananthan, P. Torr, R. Cipolla: Hirerarchical Part-Based Human Body Pose Estimation. In *British Machine Vision Conference*, pp. 949 – 958, Oxford, UK, August 2005.

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

[Neidle 02] C. Neidle: SignStream™Annotation: Conventions used for the American Sign Language Linguistic Research Project. Technical report, Boston University, August 2002.

[Neidle 07] C. Neidle: SignStream™Annotation: Addendum to Conventions used for the American Sign Language Linguistic Research Project. Technical report, Boston University, August 2007.

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

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

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

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

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

[Okuma & Taleghani+ 04] K. Okuma, A. Taleghani, N. De Freitas, J.J. Little, D.G. Lowe: A Boosted Particle Filter: Multitarget Detection And Tracking. In *European Conference on Computer Vision*, 2004.

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

[Othman & Aboulnasr 00] H. Othman, T. Aboulnasr: Low Complexity 2-D Hidden Markov Model for Face Recognition. In *IEEE International Symposium On Circuits And Systems*, Vol. 5, pp. 33–36, May 2000.

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

[Ramanan & Forsyth$^+$ 07] D. Ramanan, D.A. Forsyth, A. Zisserman: Tracking People by Learning Their Appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 1, pp. 65 – 81, January 2007.

[Ren & Berg$^+$ 05] X. Ren, A.C. Berg, J. Malik: Recovering Human Body Configurations using Pairwise Constraints between Parts. In *IEEE International Conference on Computer Vision*, 2005.

[Ronfard & Schmid$^+$ 02] R. Ronfard, C. Schmid, B. Triggs: Learning to Parse Pictures of People. In *European Conference on Computer Vision*, pp. 700–714, 2002.

[Rybach 06] D. Rybach: Appearance-Based Features For Automatic Continous Sign Language Recognition, June 2006.

[Schiele 06] B. Schiele: Model-free tracking of cars and people based on color regions. *Image and Vision Computing*, Vol. 24, pp. 1172–1178, 2006.

[Sigal & Isard$^+$ 03] L. Sigal, M. Isard, B.H. Sigelman, M.J. Black: Attractive People: Assembling Loose-Limbed Models using Non-parametrix Belief Propagation. In *Advances in Neural Information Processing Systems*, Vol. 16, pp. 1539–1546, 2003.

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

[Starner & Weaver$^+$ 98] T. Starner, J. Weaver, A. Pentland: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 20, No. 12, December 1998.

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

[Viola & Jones 04] P. Viola, M. Jones: Robust real-time face detection. *International Journal on Computer Vision*, Vol. 57, No. 2, pp. 137 – 154, 2004.

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

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

[Wang & Athitsos$^+$ 07] J. Wang, V. Athitsos, S. Sclaroff, M. Betke: Detecting Objects of Variable Structure With Hidden State Shape Models. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, Vol., pp. in press, 2007.

[Wang & Chen$^+$ 06] C. Wang, X. Chen, W. Gao: Re-sampling for Chinese Sign Language Recognition. In *Gesture in Human-Computer Interaction and Simulation*, Vol. 3881 of *Lecture Notes in Computer Science*, pp. 57–67, Feb. 2006.

[Wang & Suter$^+$ 07] H. Wang, D. Suter, K. Schindler, C. Shen: Adaptive Object Tracking Based on an Effective Appearance Filter. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 29, No. 9, pp. 1661–1667, Sept. 2007.

[Woodland 01] P.C. Woodland: Speaker Adaptation for Continuous Density HMMs: A Review, Invited Lecture. In *ITRW on Adaptation Methods for Speech Recognition*, pp. 11 – 19, August 2001.

[Wu & Nevatia 07] B. Wu, R. Nevatia: Detection And Tracking of Multiple, Partially Occluded Humans by Bayesian Combination Of Edgelet Based Part Detectors. *International Journal of Computer Vision*, Vol. 75, pp. 247 – 266, 2007.

[Zahedi 07] M. Zahedi: *Robust Appearance-based Sign Language Recognition*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, Sept. 2007.