Univ Access Inf Soc
DOI 10.1007/s10209-007-0104-x

LONG PAPER

Recent developments in visual sign language recognition

Ulrich von Agris · Jörg Zieren · Ulrich Canzler ·
Britta Bauer · Karl-Friedrich Kraiss

Abstract Research in the field of sign language recognition has made significant advances in recent years. The present achievements provide the basis for future applications with the objective of supporting the integration of deaf people into the hearing society. Translation systems, for example, could facilitate communication between deaf and hearing people in public situations. Further applications, such as user interfaces and automatic indexing of signed videos, become feasible. The current state in sign language recognition is roughly 30 years behind speech recognition, which corresponds to the gradual transition from isolated to continuous recognition for small vocabulary tasks. Research efforts were mainly focused on robust feature extraction or statistical modeling of signs. However, current recognition systems are still designed for signer-dependent operation under laboratory conditions. This paper describes a comprehensive concept for robust visual sign language recognition, which represents the recent developments in this field. The proposed recognition system aims for signer-independent operation and utilizes a single video camera for data acquisition to ensure user-friendliness. Since sign languages make use of manual and facial means of expression, both channels are employed for recognition. For mobile operation in uncontrolled environments, sophisticated algorithms were developed that robustly extract manual and facial features. The extraction of manual features relies on a multiple hypotheses tracking approach to resolve ambiguities of hand positions. For facial feature extraction, an active appearance model is applied which allows identification of areas of interest such as the eyes and mouth region. In the next processing step, a numerical description of the facial expression, head pose, line of sight, and lip outline is computed. The system employs a resolution strategy for dealing with mutual overlapping of the signer's hands and face. Classification is based on hidden Markov models which are able to compensate time and amplitude variances in the articulation of a sign. The classification stage is designed for recognition of isolated signs, as well as of continuous sign language. In the latter case, a stochastic language model can be utilized, which considers uni- and bigram probabilities of single and successive signs. For statistical modeling of reference models each sign is represented either as a whole or as a composition of smaller subunits—similar to phonemes in spoken languages. While recognition based on word models is limited to rather small vocabularies, subunit models open the door to large vocabularies. Achieving signer-independence constitutes a challenging problem, as the articulation of a sign is subject to high interpersonal variance. This problem cannot be solved by simple feature normalization and must be addressed at the classification level. Therefore, dedicated adaptation methods known from speech recognition were implemented and modified to consider the specifics of sign languages. For rapid adaptation to unknown signers the proposed recognition system employs a combined approach of maximum likelihood linear regression and maximum a posteriori estimation.

Keywords Sign language recognition · Human–computer interaction · Computer vision · Statistical pattern recognition · Hidden Markov models · Signer adaptation

U. von Agris (✉) · J. Zieren · U. Canzler · B. Bauer ·
K.-F. Kraiss
Institute of Man–Machine Interaction,
RWTH Aachen University, Ahornstrasse 55,
52074 Aachen, Germany
e-mail: vonagris@mmi.rwth-aachen.de

1 Introduction

Sign language is a non-verbal language used by deaf and hard of hearing people for everyday communication among themselves. Information is conveyed visually, using a combination of manual and non-manual means of expression. The manual parameters are hand shape, hand posture, hand location, and hand motion. The non-manual parameters include head and body posture, facial expression, gaze and mouth movements. The latter encode, e.g., adjectives and adverbials, contribute to grammar or provide specialization of general items.

Some signs can be distinguished by manual parameters alone, while others remain ambiguous unless additional non-manual information is made available. Unlike pantomime, sign language does not include its environment. Signing takes place in a three-dimensional space close to the signer's trunk and head, called signing space. Signs are performed either one-handed or two-handed. For one-handed signs the action of only one hand is required, where a person generally uses the same hand, known as the dominant hand.

The grammar of sign language is fundamentally different from spoken language. The structure of a sentence in spoken language is linear, one word followed by another, whereas in sign language, a simultaneous structure exists with a parallel temporal and spatial configuration. The configuration of a sign language sentence carries rich information about time, location, person, or predicate.

Spread all over the world, sign language is not universal. Nationally different languages have evolved, such as German Sign Language (DGS) or American Sign Language (ASL). Just like in spoken language, there are regional dialects in sign language. In contrast to the pronunciation of words, however, there is no standard for signs, and people may use an altogether different sign for the same word. Even when performing identical signs, the variations between different signers are considerable.

1.1 Applications for sign language recognition

Unfortunately, very few hearing people are able to communicate in sign language. The use of interpreters is often prohibited by limited availability and high cost. This leads to problems in the integration of deaf people into society, and conflicts with an independent and self-determined lifestyle. For example, many deaf people are unable to use the World Wide Web and communicate by e-mail in the way hearing people do, since they commonly have great difficulties in reading and writing. The reason for this is that hearing people learn and perceive written language as a visual representation of spoken language. For deaf people, however, this correspondence does not exist, and letters—which encode phonemes—are just symbols without any meaning.

In order to improve communication between deaf and hearing people, research in automatic sign language recognition is needed. This work shall provide the technical requirements for translation systems and user interfaces that support the integration of deaf people into the hearing society. The aim is the development of a mobile system, consisting of a laptop equipped with a webcam that visually reads a signer's gestures and facial expression, and performs a translation into spoken language. This device is intended as an interpreter in everyday life, e.g., at the post office or in a bank. Furthermore, it allows deaf people intuitive access to electronic media such as computers or the Internet (Fig. 1).

Regarding the Internet, a barrier-free access also includes technical facilities to search the Web for information in sign language. Current search engines solely conduct a textual analysis of websites, simply ignoring the visual information contained in signed videos. Although the number of videos is increasing, their contents cannot be searched and retrieved like text due to missing automatic indexing methods. This creates difficulties for deaf people to gather information on their own. In this context, sign language recognition could be utilized for the indexing task, providing the required basis for sign language search engines.

Another envisaged application is the development of a sign language tutor used in the areas of both education and rehabilitation. It could support patients suffering from hearing loss, deaf people with sign language deficiencies, as well as interested hearing people, in learning sign language. The user is first presented with the sign to be learnt by means of a video. He then performs the sign himself, imitating the signer on the screen. Subsequently, the user's

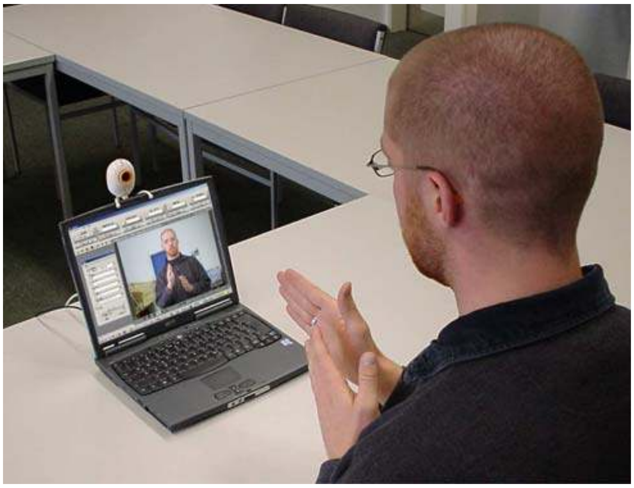


Fig. 1 Laptop and webcam-based sign language recognition system

signing is analyzed by the tutoring system which provides feedback regarding the correctness of execution, such as possibly required modifications of hand position and posture.

Many further applications arise both inside and outside the field of sign language. Research on human–computer interaction could also benefit from gesture and mimic analysis algorithms, originally developed for sign language recognition systems. However, all mentioned applications have in common that they must ensure highest usability and user-friendliness. Visual non-intrusive approaches are generally most suited to meet these requirements. Furthermore, the recognition system should allow user-independent operation in an uncontrolled environment.

1.2 State of the art in sign language recognition

The current state in automatic sign language recognition is roughly 30 years behind speech recognition due to manifold reasons. Processing and classification of two-dimensional video signals are significantly more complex than of one-dimensional audio signals. In addition, sign language is by far not fully explored yet. Little is known about syntax and semantics, and no dictionary exists. The lack of national media—such as radio, TV, and telephone for the hearing—leads to strong regional variation. For a large number of signs there is not even a common definition.

Sign language recognition has become the subject of scientific publications only in the beginning of the 90s. Most presented systems operate near real-time and require up to 10 s of processing time after completion of the sign. For video-based approaches, details on camera hardware and resolution are rarely published, suggesting that professional equipment, high resolution, low noise, and optimal camera placement was used.

The method of data acquisition defines a user interface's quality and constitutes the primary feature for classification of different works. The most reliable, exact, and at the same time the simplest techniques are intrusive. Data gloves measure the flexion of the finger joints, optical or magnetic markers placed on face and hands facilitate a straightforward determination of facial expression and manual configuration. For the user, however, this is unnatural and restrictive. Furthermore, data gloves are unsuitable for practical applications due to their high cost. Also most existing systems exploit manual features only; so far facial features were rarely used [30].

Usability of video-based recognition systems is greatly influenced by the robustness of its image processing stage, i.e., its ability to handle inhomogeneous, dynamic, or generally uncontrolled backgrounds and suboptimal illumination. Many publications do not explicitly address this issue, which—in connection with accordant illustration—suggests homogeneous backgrounds and strong diffuse lighting. Another common assumption is that the signer wears long-sleeved clothing that differs in color from his skin, allowing color-based detection of hands and face.

The majority of systems only support signer-dependent operation, i.e., every user is required to train the system before being able to use it. Signer-independent operation requires a suitable normalization of features early in the processing chain to eliminate dependencies of the features on the signer's position in the image, his distance from the camera, and the camera's resolution. This is rarely described in publications; instead, areas and distances are measured in pixels, which even for signer-dependent operation would require an exact reproduction of the conditions under which the training material was recorded.

Similar to the early days of speech recognition, most researchers focus on isolated signs. While several systems exist that process continuous signing, their vocabulary is very small. Recognition rates of 90% and higher are reported, but the exploitation of context and grammar—which is sometimes rigidly fixed to a certain sentence structure—aid considerably in classification. As in speech recognition, coarticulation effects and the resulting ambiguities form the primary problem when using large vocabularies.

Table 1 lists several important publications and the described systems' features. When comparing the indicated performances, it must be kept in mind that, in contrast to speech recognition, there is no standardized benchmark for sign language recognition. Thus, recognition rates cannot be compared directly. The compilation also shows that most systems' vocabularies are in the range of 50 signs.

Larger vocabularies have only been realized with the use of data gloves. All systems are signer-dependent. All recognition rates are valid only for the actual test scenario. Information about robustness in real-life settings is not available for any of the systems. Furthermore, the exact constellation of the vocabularies is unknown, despite its significant influence on the difficulty of the recognition task. In summary, it can be stated that none of the systems currently found in literature meets the requirements for a robust real world application.

1.3 Visual sign language recognition

This paper outlines the implementation of an existing visual sign language recognition system for real world applications, which surpasses the current state of the art in many respects. The reader interested in a more detailed

Table 1 Classifier characteristics for user dependent sign language recognition

Author, Year	Features	Interface	Vocabulary	Language Level	Recognition rate in %
Vamplew, 1996 [40]	Manual	Data glove	52	Word	94.0
Holden, 2001 [15]	Manual	Optical markers	22	Word	95.5
Yang, 2002 [46]	Manual	Video	40	Word	98.1
Murakami, 1991 [27]	Manual	Data glove	10	Sentence	96.0
Liang, 1997 [24]	Manual	Data glove	250	Sentence	89.4
Fang, 2002 [12]	Manual	Data glove	203	Sentence	92.1
Starner, 1998 [35]	Manual	Video	40	Sentence	97.8
Vogler, 1999 [42]	Manual	Video	22	Sentence	91.8
Parashar, 2003 [30]	Manual and facial	Video	39	Sentence	92.0

description of this recognition system or an in-depth introduction to gesture and sign language recognition is directed to [21]. Figure 2 shows a schematic of the process, which can be divided into a feature extraction stage and a subsequent classification stage.

The recognition system utilizes a single video camera for data aquisition. The input image sequence is forwarded to two parallel processing chains that extract manual and facial features using a priori knowledge of the signing process. Before the final classification is performed, a pre-classification module restricts the active vocabulary to reduce processing time. Manual and facial features are then classified separately, and both results are merged to yield a single recognition result.

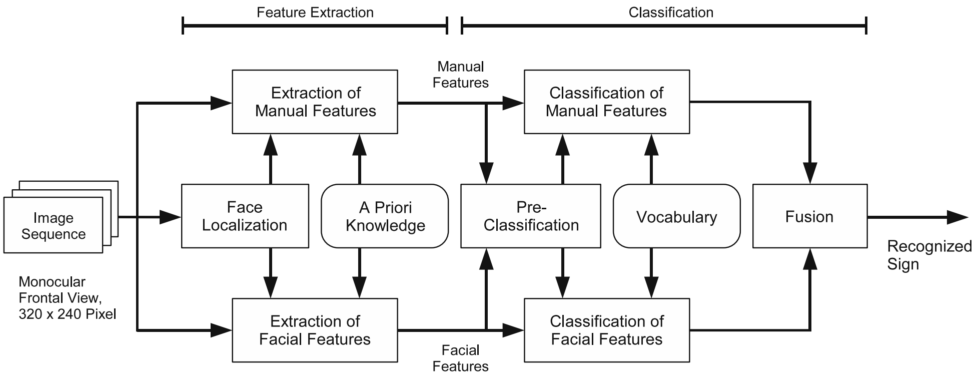
This paper is structured as follows: Sect. 2 describes the algorithms used for robust extraction of manual features in uncontrolled environments. Section 3 explains how facial features can be integrated into the sign language recognition process. The extracted features comprise facial expression, head pose, line of sight, and lip outline. Section 4 deals with the classification of isolated signs and of continuous signing. Two approaches for statistical modeling of reference models will be introduced: word models for small vocabularies and subunit models for large vocabularies. Section 5 addresses the problem of high interpersonal variance in the articulation of a sign. As a solution, dedicated adaptation methods known from speech recognition are applied for rapid adaptation to unknown signers. Finally, Sect. 6 presents some performance evaluations for the presented recognition system.

2 Extraction of manual features

Sign language recognition constitutes a challenging field of research in computer vision. Compared to gesture recognition in controlled environments, recognition of sign language in real world scenarios places significantly higher demands on feature extraction and processing algorithms. With regard to the two-dimensional input data, the following problems arise:

- While most gestures are one-handed, signs may be one- or two-handed. The system must therefore not only be able to handle two moving objects, but also to detect whether the non-dominant hand is idle or move together with the dominant hand.
- Since the hands' position relative to the face carries a significant amount of information in sign language, the face must be included in the image as a reference point.

Fig. 2 Schematic of the sign language recognition system


Feature Extraction
Classification
Manual Features
Extraction of Manual Features
Classification of Manual Features
Image Sequence
Face Localization
A Priori Knowledge
Pre-Classification
Vocabulary
Fusion
Recognized Sign
Monocular Frontal View, 320 x 240 Pixel
Extraction of Facial Features
Classification of Facial Features
Facial Features

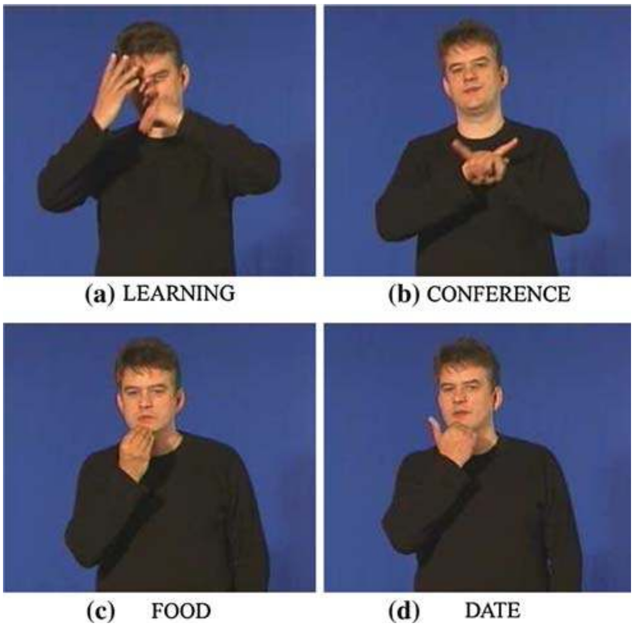


Fig. 3 Difficulties in sign language recognition: overlap of hands and face (**a**), tracking of both hands in ambiguous situations (**b**), and similar signs with different meaning (**c**, **d**)

This reduces image resolution available for the hands and poses an additional localization task.

- Hands may occlude one another and/or the face (Fig. 3a, b). Because of the two-dimensional projection in the image plane, these objects appear as a single object. A reliable segmentation of the individual objects is not possible, resulting in a loss of information.
- When using color and/or motion to detect the user's hands and face, uncontrolled backgrounds as shown in Fig. 4 may give rise to numerous false alarms that require high-level reasoning to tell targets from distractors.
- Signs are often very similar (or even identical) in their manual features and differ mainly (or exclusively) in non-manual features (Fig. 3c, d). This makes automatic recognition based on manual features difficult or, for manually identical signs, even unfeasible.

Obviously, the segmentation of the signer's hands and face are computationally expensive and error-prone, because the required knowledge and experience that comes natural to humans is difficult to encode in machine-readable form. A multitude of algorithms has been published that aim at the solution of the above problems [10, 29, 47].

2.1 Global hand features: geometric features

This subsection focuses on the robust extraction of geometric features describing the two-dimensional projection of a hand in the image plane. The feature extraction stage is extended as shown in Fig. 5. An image preprocessing stage is added that applies low-level algorithms for image enhancement, such as background modeling. High-level knowledge is applied for the resolution of overlaps and to support hand localization and tracking.

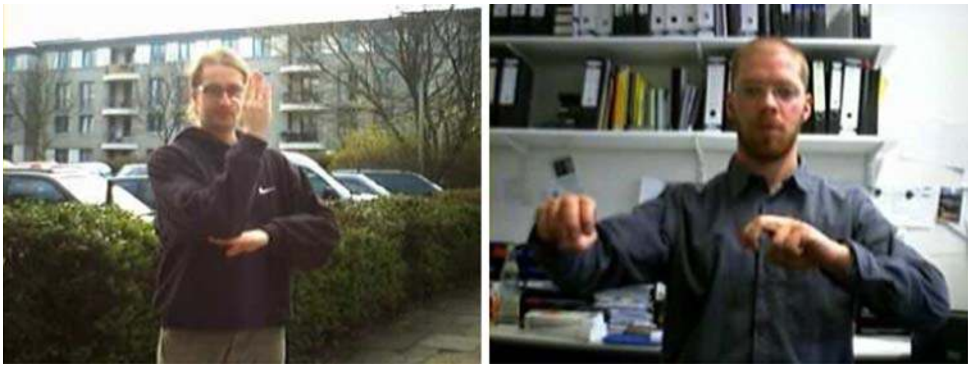
2.1.1 Image preprocessing

The preprocessing stage improves the quality of the input images to increase the performance (in terms of processing speed and accuracy) of the subsequent stages. High-level information computed in those stages may be exploited, but this bears the usual risks of a feedback loop, such as instability or the reinforcement of errors. Low-level algorithms, on the other hand, do not have this problem and can often be used in a wide range of applications. A prominent example is the modeling of the image background.

Background subtraction The detection of static background areas in dynamic scenes is an important step in many pattern recognition systems. Background subtraction, the exclusion of these areas from further processing, can significantly reduce clutter in the input images and decrease the amount of data to be processed in subsequent stages.

If a view of the entire background without any foreground objects is available, a statistical model can be

Fig. 4 Examples for uncontrolled backgrounds that may interfere with the tracking of hands and face



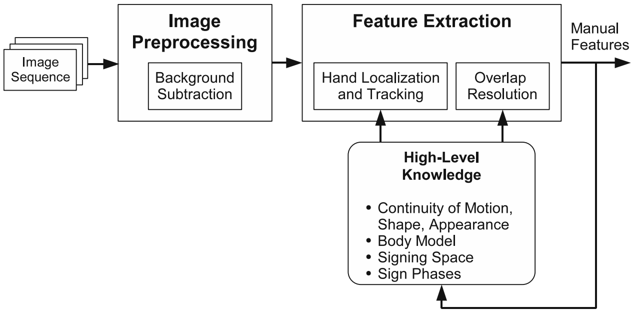


Fig. 5 Feature extraction extended by an image preprocessing stage. High-level knowledge of the signing process is used for overlap resolution and hand localization/tracking

calculated in a calibration phase. This approach is rather impracticable in particular for mobile applications. In the following, it is thus assumed that the only available input data is the signing user, and that the signing process takes about 1–10 s. Therefore, the background model is to be created directly from the corresponding 25–250 frames (assuming 25 fps) containing both background and foreground, without prior calibration.

Median background model A simple yet effective method to create a background model in the form of an image $\mathbf{I}_{\mathrm{bg}}(x,y)$ is to compute, for every pixel (x,y), the median color over all frames $\mathbf{I}$(x,y,t):

$$\mathbf{I}_{\mathrm{bg}}(x,y) = \mathrm{median}\{\mathbf{I}(x,y,t) | 1 \le t \le T\} \tag{1}$$

where the median of a set of vectors $V = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ is the element for which the sum of Euclidian distances to all other elements is minimal:

$$\mathrm{median} V = \underset{\mathbf{v}\in V}{\mathrm{argmin}} \sum_{i=1}^{n} |\mathbf{v}_i - \mathbf{v}| \tag{2}$$

When using RGB color space, each vector $\mathbf{v}_i$ represents the color of a single pixel as a 3-tuple (r, g, b) of scalar values specifying the color's red, green, and blue components.

For the one-dimensional case the median is equivalent to the 50th percentile, which is significantly faster to compute, e.g., by simply sorting the elements of V. Therefore, (2) is in practice often approximated by the channel-wise median

$$\mathbf{I}_{\mathrm{bg}}(x,y) = \begin{pmatrix} \mathrm{median}\{r(x,y,t)|1 \le t \le T\} \\ \mathrm{median}\{g(x,y,t)|1 \le t \le T\} \\ \mathrm{median}\{b(x,y,t)|1 \le t \le T\} \end{pmatrix} \tag{3}$$

The median background model has the convenient property of not requiring any parameters, and is thus very robust. Its only requirement is that the image background must be visible at the considered pixel in more than 50% of the input frames, which is a reasonable assumption in most scenarios. A slight drawback might be that all input frames need to be buffered in order to compute (1).

The application of the background model to a frame, i.e., classification of a given pixel as background or foreground, requires the definition of a suitable metric Δ to quantify the difference between a background color vector $(r_{\mathrm{bg}}\ g_{\mathrm{bg}}\ b_{\mathrm{bg}})^{\mathrm{T}}$ and a given color vector $(r\ g\ b)^{\mathrm{T}}$:

$$\Delta\left((r\ g\ b)^{\mathrm{T}}, (r_{\mathrm{bg}}\ g_{\mathrm{bg}}\ b_{\mathrm{bg}})^{\mathrm{T}}\right) \ge 0 \tag{4}$$

A sufficient implementation of Δ is the Euclidian distance. Computing Δ for every pixel in an input image $\mathbf{I}(x,\ y,\ t)$ and comparison with a motion sensitivity threshold Θ_{m} yields a foreground mask $\mathbf{I}_{\mathrm{fg,mask}}$:

$$\mathbf{I}_{\mathrm{fg,mask}}(x,y,t) = \begin{cases} 1 & \text{if} \quad \Delta\left(\mathbf{I}(x,y,t), \mathbf{I}_{\mathrm{bg}}(x,y)\right) \ge \Theta_{\mathrm{m}} \\ 0 & \qquad \text{otherwise} \end{cases} \tag{5}$$

Here Θ_{m} is chosen just large enough so that the camera noise is not classified as foreground. Thus, while the computation of the median background model is itself parameter-free, its application involves the parameter Θ_{m}.

Figure 6 shows four example images from an input video sequence of 54 frames in total. Each frame of the sequence contains at least one person in the background. The resulting background model, as well as an exemplary application to a single frame, is visualized in Fig. 7.

2.1.2 Feature extraction

Recognition systems that must operate in real world conditions require sophisticated feature extraction approaches. The extraction stage aims for the robust segmentation of the signer's hands and face (which is needed as a reference point) from the input image sequence. Since a sign may be one- or two-handed, it is impossible to know in advance whether the non-dominant hand will remain idle or move together with the dominant hand. Background subtraction alone cannot be expected to isolate foreground objects without errors or false alarms. Also, the face—which is mostly static—has to be localized before background subtraction can be applied.

Since the hands' extremely variable appearance prevents the use of shape or texture cues, it is common to exploit only color for hand localization. This leads to the restriction that the signer must wear long-sleeved, non-skin-colored clothing to facilitate the separation of the hand from the arm by means of color. By using a generic skin color model such as the one presented in [20], user and illumination independence can be achieved at the cost of a high number of false alarms. This method therefore requires high-level reasoning algorithms to handle these ambiguities.

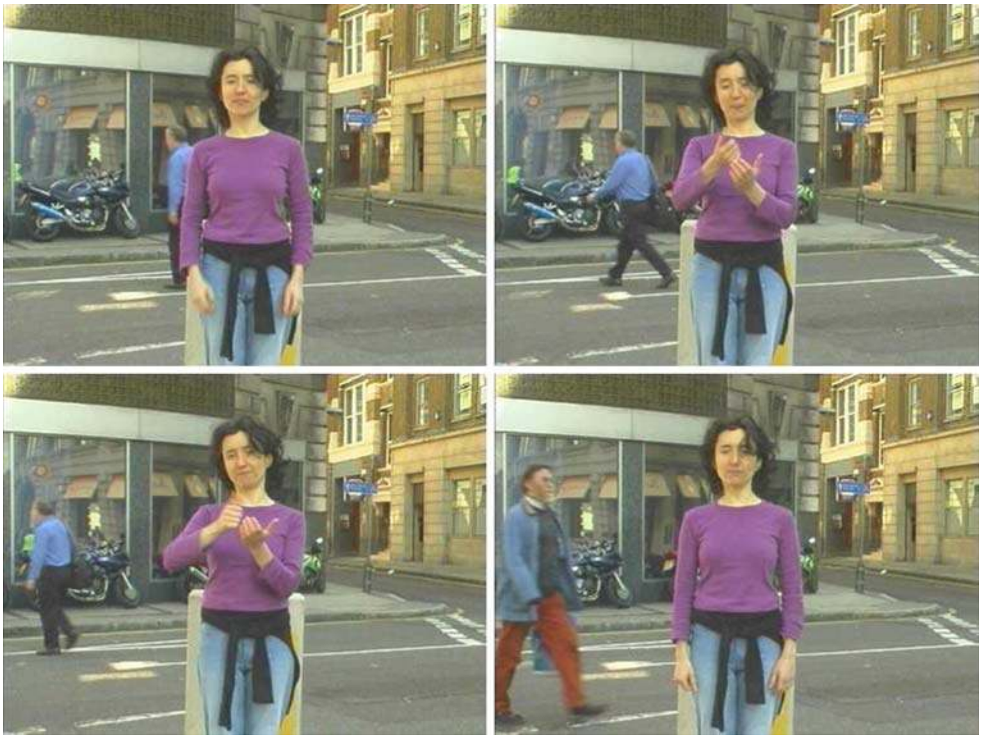
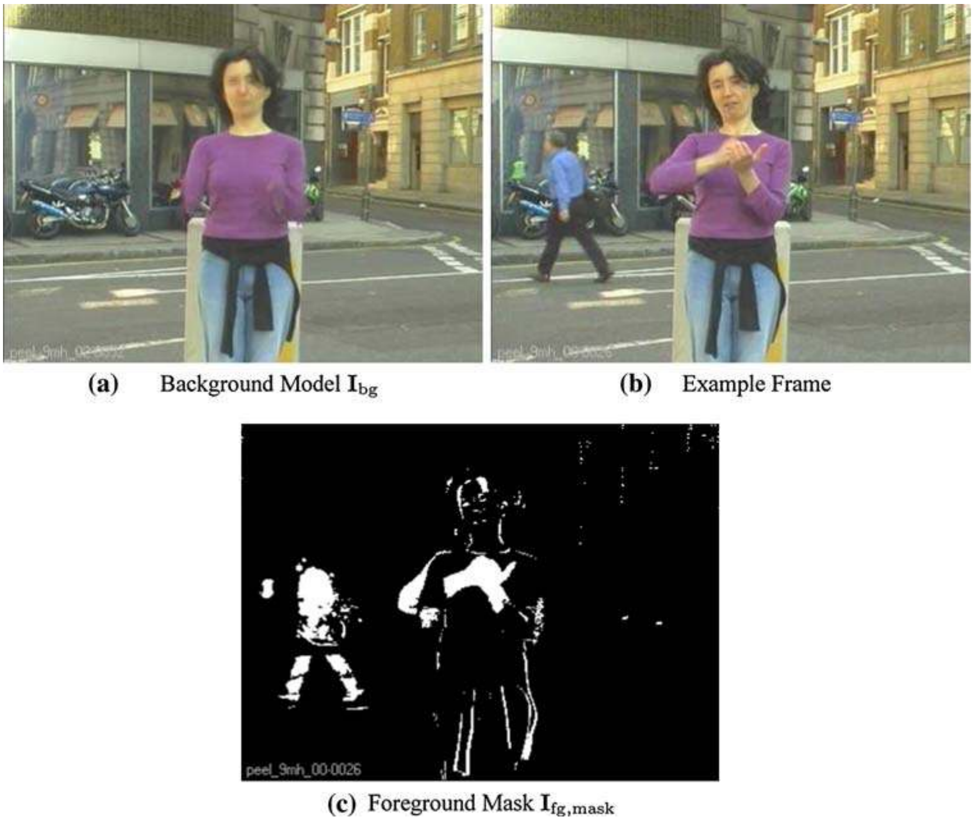


Fig. 6 Four example frames from the British sign PEEL. The input video consists of 54 frames, all featuring at least one person in the background



(a) Background Model $\mathbf{I}_{\text{bg}}$

(b) Example Frame

(c) Foreground Mask $\mathbf{I}_{\text{fg,mask}}$

Fig. 7 Background model computed for the input shown in Fig. 6 (**a**), its application to an example frame (**b**), and the resulting foreground mask (**c**)

2.1.2.1 Hand tracking Performing simple hand localization in every frame is not sufficiently reliable in complex scenarios. The relation between temporally adjacent frames has to be exploited in order to increase performance. Localization is thus replaced by tracking, using information from previous frames as a basis for finding the hand in the current frame.

A common problem in hand tracking is the handling of ambiguous situations where more than one interpretation is plausible. For instance, Fig. 8a shows the skin color segmentation of a typical scene (Fig. 7b). This observation does not allow a direct conclusion as to the actual hand configuration. Instead, there are multiple interpretations, or hypotheses, as visualized in Fig. 8b, c, d.

Simple approaches weigh all hypotheses against each other and choose the most likely one in every frame on the basis of information gathered in preceding frames, such as, for instance, a position prediction. All other hypotheses are discarded. This concept is error-prone because ambiguous situations that cannot reliably be interpreted occur frequently in sign language. Robustness can be increased significantly by evaluating not only preceding, but also subsequent frames, before any hypothesis is discarded. It is therefore desirable to delay the final decision on the hands' position in each frame until all available data has been analyzed and the maximum amount of information is available.

These considerations lead to the implementation of a multiple hypotheses tracking (MHT) approach [47]. In a first pass of the input data, all conceivable hypotheses are created for every frame. Transitions are possible from each hypothesis at time t to all hypotheses at time $t+1$, resulting in a state space as shown exemplarily in Fig. 9. The total number of paths through this state space equals $\prod_t H(t)$, where $H(t)$ denotes the number of hypotheses at time t. Provided that the skin color segmentation detected both hands and the face in every frame, one of these paths represents the correct tracking result. In order to find this path (or one as close as possible to it), probabilities are computed that indicate the likeliness of each hypothesized configuration, p_{state}, and the likeliness of each transition, $p_{\text{transition}}$ (see Fig. 9).

The computation of p_{state} and $p_{\text{transition}}$ is based on high-level knowledge, encoded as a set of rules or learned in a training phase. The following aspects, possibly extended by application-dependent items, provide a general basis:

- The physical configuration of the signer's body can be deduced from position and size of face and hands. Configurations that are anatomically unlikely or do not occur in sign language reduce p_{state}.
- The three phases of a sign (preparation, stroke, retraction), in connection with the signer's handedness, should also be reflected in the computation of p_{state}. The handedness has to be known in order to correctly interpret the resulting feature vector.
- Even in fast motion, the hand's shape changes little between successive frames at 25 fps. As long as no

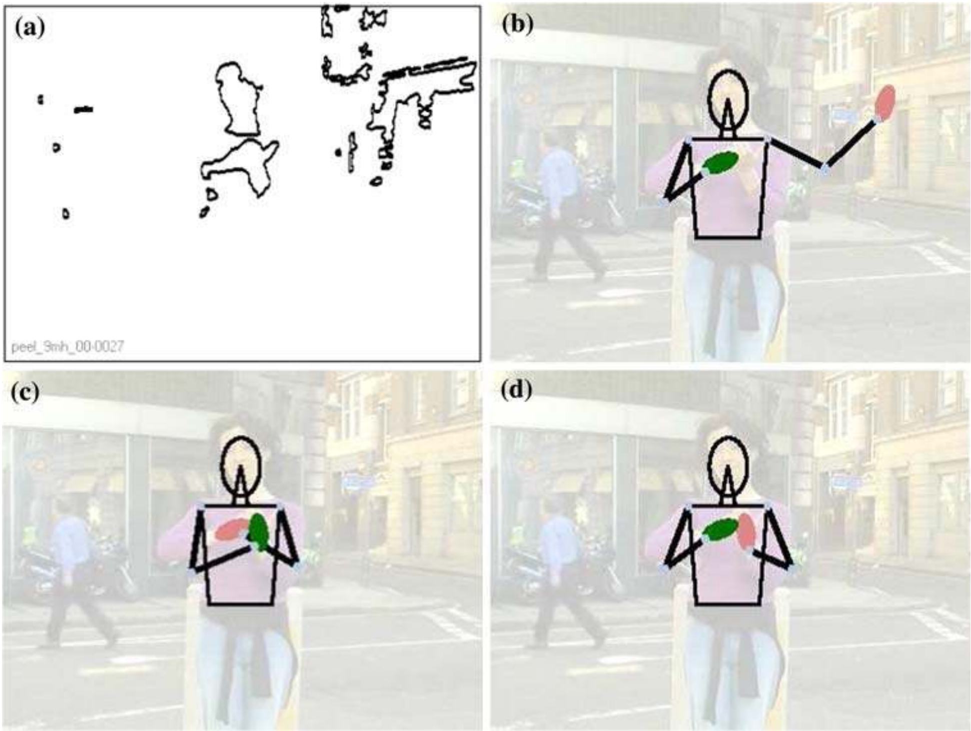


Fig. 8 Skin color segmentation of Fig. 7b (**a**) and a subset of the corresponding hypotheses (**b**, **c**, **d**; correct: **d**)

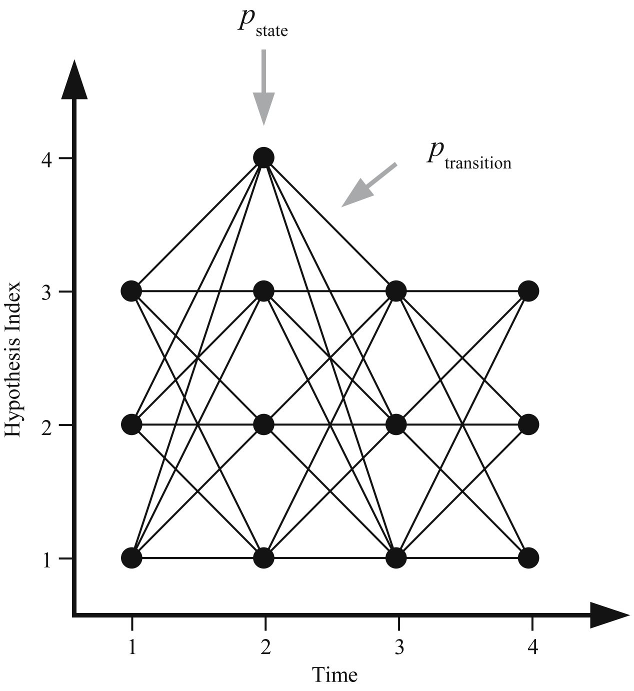


Fig. 9 Hypothesis space and probabilities for states and transitions

overlap occurs, the shape at time t can therefore serve as an estimate for time $t + 1$. With increasing deviation of the actual from the expected shape, $p_{\text{transition}}$ is reduced. Abrupt shape changes due to overlap require special handling.

- Similarly, hand position changes little from frame to frame (at 25 fps), so that coordinates at time t may serve as a prediction for time $t + 1$. Kalman filters [45] may increase prediction accuracy by extrapolating on the basis of all past measurements.
- Keeping track of the hand's mean or median color can prevent confusion of the hand with nearby distractors of similar size but different color. This criterion affects $p_{\text{transition}}$.

For searching the hypothesis space, the Viterbi algorithm is applied in conjunction with pruning of unlikely paths [33].

The MHT approach ensures that all available information is evaluated before the final tracking result is determined. The tracking stage can thus exploit, at time t, information that becomes available only at time $t_1 > t$. Errors are corrected retrospectively as soon as they become apparent.

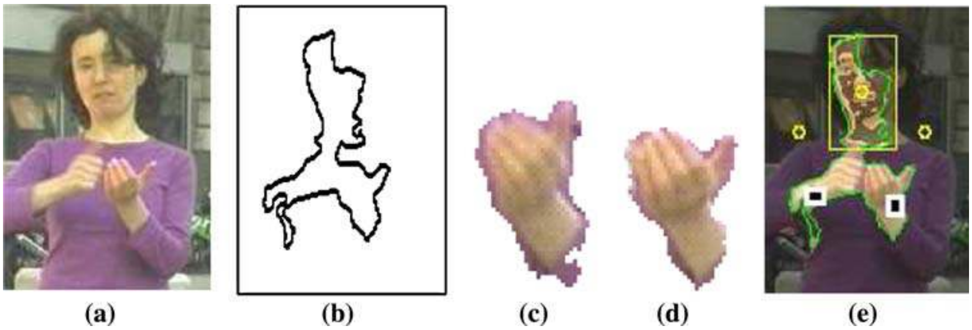
2.1.2.2 Overlap resolution In numerous signs both hands overlap with each other and/or with the face. When two or more objects overlap in the image, the skin color segmentation yields only a single blob for all of them, rendering a direct extraction of meaningful features impossible (Fig. 10). Low contrast, low resolution, and the hands' variable appearance usually do not allow a separation of the overlapping objects by an edge-based segmentation either. Most of the geometric features available for not overlapped objects can therefore not be computed for overlapping objects, and have to be interpolated.

However, the hand's appearance is sufficiently constant over several frames for template matching [34] to be applied. Using the last not overlapped view of each overlapping object as a template, at least position features—which carry much information—can be reliably computed during overlap. The accuracy of this method decreases with increasing template age. Fortunately, the same technique can be used twice for a single period of overlap, the second time starting with the first not overlapped view after the cessation of the overlap, and proceeding temporally backwards. This effectively halves the maximum template age and increases precision considerably.

2.1.3 Feature computation

The subsequent classification stage requires a numerical description of the signer's hand configuration in each frame. For this purpose, several geometric features are computed from the hand candidate border, as shown in Fig. 11. These features, describing only the two-dimensional projection of each hand in the image, plane are:

Fig. 10 Template matching for overlap resolution: input image (**a**), skin color segmentation (**b**), last view before overlap (**c**), first view after overlap (**d**), and estimated hand centers (**e**)



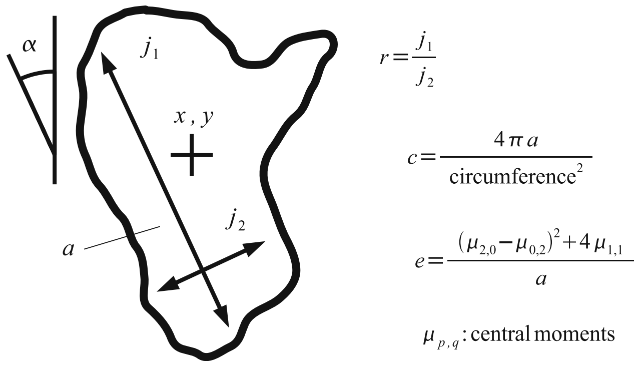

α
j_1
x, y
j_2
a
$r = \frac{j_1}{j_2}$
$c = \frac{4\pi a}{\text{circumference}^2}$
$e = \frac{(\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}}{a}$
$\mu_{p,q}$: central moments

Fig. 11 Geometric features computed for each hand

- Center coordinates x, y
- Area a
- Orientation α of main axis
- Ratio r of inertia along/perpendicular to main axis
- Compactness c
- Eccentricity e

Since $\alpha \in [-90°, 90°]$, the orientation is split into $o_1 = \sin 2\alpha$ and $o_2 = \cos \alpha$ to ensure stability at the interval borders. The derivatives $\dot{x}, \dot{y}$, and $\dot{a}$ complete the 22-dimensional feature vector, which combines the features of both hands:

$$x_t = [\underbrace{x\,\dot{x}\,y\,\dot{y}\,a\,\dot{a}\,o_1\,o_2\,r\,c\,e}_{\text{left hand}}\quad\underbrace{x\,\dot{x}\,y\,\dot{y}\ldots}_{\text{right hand}}] \tag{6}$$

During periods of overlap, template matching is performed to accurately determine the center coordinates x, y using preceding or subsequent not overlapped views. All other features are linearly interpolated. If the hand is not visible or remains static throughout the sign, its features are set to zero.

For signer-independent mobile application, some of the above features need to be normalized. For example, the area a depends on image resolution and on the signer's distance to the camera. The hand coordinates x, y additionally depend on the signer's position in the image. Using the face position and size as a reference for normalization can eliminate both translation and scale variance. The center coordinates x, y are thus specified relative to the corresponding shoulder position, which is estimated from the width w_F and position of the face. In addition, x,y are normalized by w_F, and a by w_F^2.

2.2 Detailed hand features: hand posture recognition

Visual feature extraction of an unmarked hand constitutes a challenging problem. A robust feature which can be extracted using a simple skin color model is the hand's contour [18, 41]. However, the contour is not stable because small changes in hand posture may greatly affect it. At the same time, discarding texture entails yet another loss of input information in addition to the 3D-to-2D projection. Many different hand postures result in the same contour (e.g., a fist and a pointing index finger seen from the pointing direction), rendering this feature problematic for unrestricted posture recognition from arbitrary viewing angles.

This section presents a very promising approach for hand posture recognition in monocular image sequences that allows measurement of joint angles, viewing angle, and position in space [11, 48]. This model-based method imposes no posture restrictions and requires no initialization or signer-dependent training. Because this approach has been recently developed, it is not fully evaluated in combination with the sign language recognition system described in this paper. However, first experimental results for hand posture recognition are provided at the end of this section.

2.2.1 *System overview*

Figure 12 shows an overview of the system. The user wears a cotton glove equipped with six differently colored visual markers, five covering approximately half of each finger and the thumb, and another on the back of the hand. This allows extracting descriptive and stable 2D features from a monocular view. The markers' geometry lends itself to an elliptical approximation in the image plane, resulting in a very compact representation. Hand posture recognition is performed by matching a synthetic hand model featuring identical markers to minimize deviation in feature space.

In an offline preparation phase the hand model is used to generate a large number of postures seen from many different view angles. All postures are stored in an appearance database in which each entry contains the 3D hand posture parameters along with the 2D features that were extracted from the corresponding synthetic image. For recognition a database with 2.6 million entries is generated. This does not include rotation in the image plane, which is computed online.

Hand posture recognition can now be performed by using the 2D features extracted from the input images as a key for querying the database. Because feature extraction relies on a color-based segmentation to detect the markers, real-life images may yield several candidates per marker. Therefore, for each frame, a fixed number of N postures whose features have high similarity to the extracted features are retrieved. For resolving ambiguities, multiple hypotheses are pursued in parallel over time, computing plausibility scores based on the candidates' geometry and

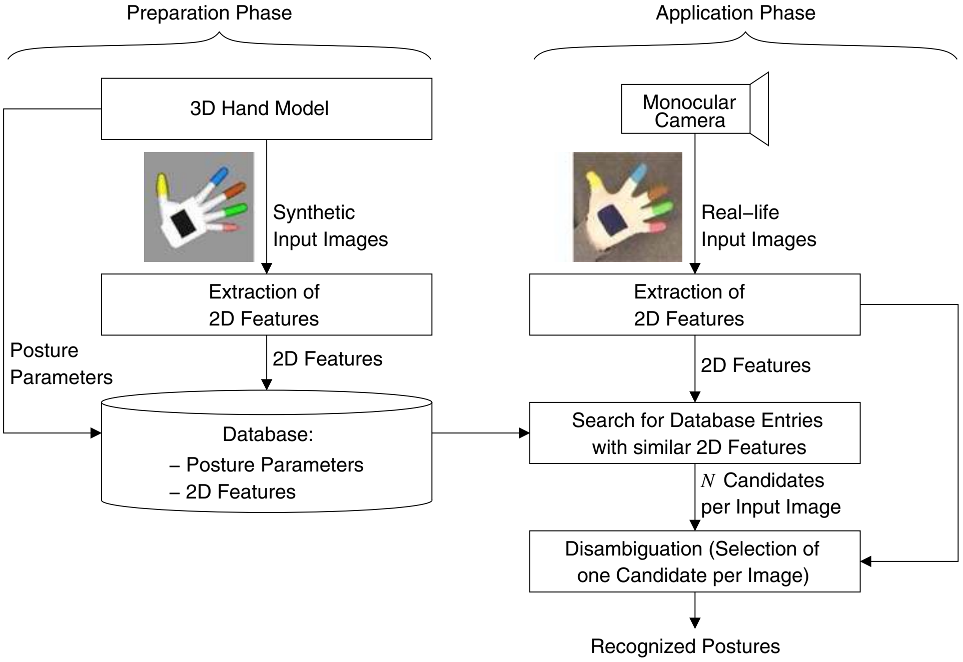


Fig. 12 System overview for extraction of detailed hand features

their continuity in feature space. The winner hypothesis is chosen at the end of the sequence, exploiting all available information. Spline interpolation between successive frames, considering match quality in each, finally yields a smooth posture sequence not restricted to the discretized posture space of the database.

2.2.2 *Hand model*

Regarding possible configurations of fingers and thumb, the human hand has 21 degrees of freedom (DOF) [37]. Each finger has one DOF for each of its joints, plus a forth DOF for sidewise abduction. The thumb requires five DOF due to its greater flexibility. The hand model reduces this to seven DOF by assuming dependencies between a finger's joints. Indices to little finger are modeled by a single parameter each, ranging from 0.0 (fully outstretched) to 1.0 (maximum bending). The thumb is modeled similarly, using two additional parameters to reflect its flexibility.

Besides dealing with finger bendings the model also handles a posture's viewing angle, i.e., the hand's orientation in space. On the surface of an imaginary sphere around the hand (called the view sphere), each point corresponds to a specific view onto the hand. A view point is therefore characterized by latitude and longitude. Additionally, for each view point a camera (or hand) rotation is possible.

2.2.3 *Experimental results*

Real-life performance has been tested on sequences of signed numbers. Figure 13 depicts some examples, each showing a magnification of the actual input image and the recognized posture. By visual comparison match quality is

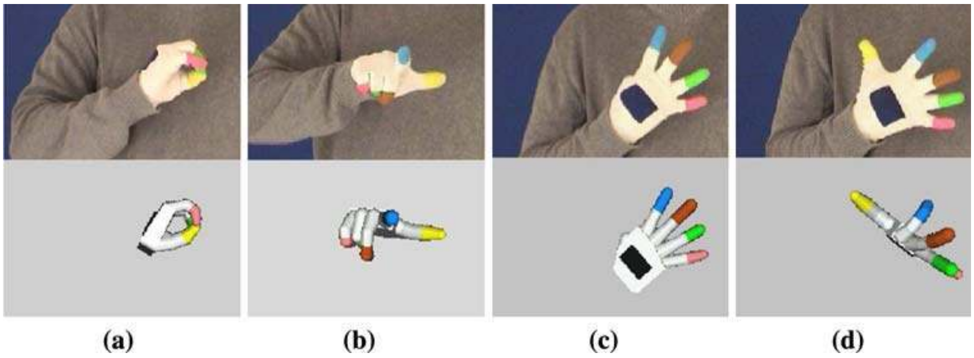


Fig. 13 Real-life examples showing four different signs (**a**–**d**). For **d** the back-of-hand marker has been removed manually

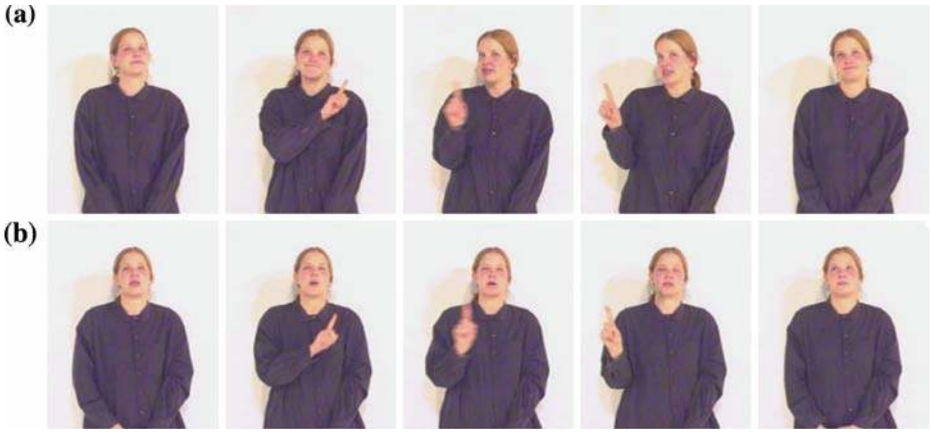


Fig. 14 The German signs NOT (NICHT) (**a**) and TO (BIS) (**b**) are identical with respect to manual gesturing but vary in head movement

high. Figure 13d illustrates the system's reaction to marker detection failures. Using a standard PC with a 1.6 GHz CPU and 1.25 GB RAM, processing speed is approximately 5 fps.

3 Extraction of facial features

Since sign languages are multimodal languages, several channels for transferring information are used at the same time. One basically differentiates between the manual/gestical channels and the non-manual/facial channels and their respective parameters [5]. In the following, first the non-manual parameters are described in more detail, and afterwards the extraction of facial features for sign language recognition is presented.

3.1 Non-manual parameters

Non-manual parameters are indispensable in signs language. They encode e.g. adjectives and adverbials and contribute to grammar. In particular, some signs are identical with respect to gesturing and can only be differentiated by making reference to non-manual parameters [6]. This is, e.g., the case for the signs NOT and TO in German Sign Language, which can only be distinguished by making reference to head motion (Fig. 14). Similarly, in British Sign Language (BSL) the signs NOW and TODAY need lip outline for disambiguation (Fig. 15). In the following the most important non-manual parameters will be described in more detail.

Upper body posture The torso generally serves as reference of the signing space. Spatial distances and textual aspects can be communicated by the posture of the torso. The signs REJECTING or ENTICING, e.g., show a slight inclination of the torso towards the rear and in forward direction. Likewise, grammatical aspects as, e.g., indirect speech, can be coded by torso posture.

Head pose The head pose also supports the semantics of sign language. For example, questions, affirmations, denials, and conditional clauses are communicated with the help of head pose. In addition, information concerning time can be coded. Signs which refer to a short time lapse are, e.g., characterized by a minimal change of head pose, while signs referring to a long lapse are performed by turning the head clearly into the direction opposite to the gesture.

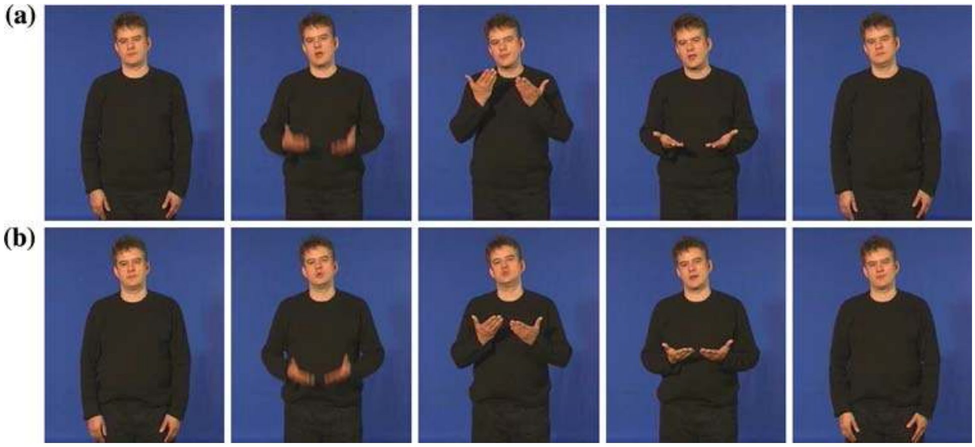


Fig. 15 The British signs NOW (**a**) and TODAY (**b**) are identical with respect to manual gesturing but vary in lip outline

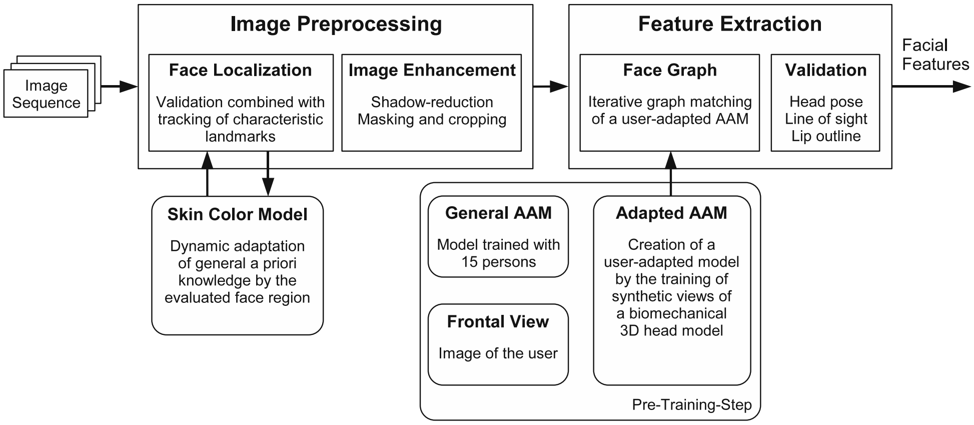


Image Sequence
Image Preprocessing
Face Localization
Validation combined with tracking of characteristic landmarks
Image Enhancement
Shadow-reduction Masking and cropping
Feature Extraction
Face Graph
Iterative graph matching of a user-adapted AAM
Validation
Head pose Line of sight Lip outline
Facial Features
Skin Color Model
Dynamic adaptation of general a priori knowledge by the evaluated face region
General AAM
Model trained with 15 persons
Frontal View
Image of the user
Adapted AAM
Creation of a user-adapted model by the training of synthetic views of a biomechanical 3D head model
Pre-Training-Step

Fig. 16 Processing chain for facial feature extraction

Line of sight Two communicating deaf persons usually establish a close visual contact. However, a brief change of line of sight can be used to refer to the spatial meaning of a gesture. In combination with torso posture, line of sight can also be used to express indirect speech, e.g., by re-enacting a conversation between two absent persons.

Facial expression Facial expressions essentially serve the transmission of feelings (lexical mimics). In addition, grammatical aspects may be encoded as well. A change of head pose combined with the lifting of the eye brows corresponds, e.g., to a subjunctive.

Lip outline Lip outline represents the most pronounced non-manual characteristic. Often it differs from voicelessly expressed words in that part of a word is shortened. Lip outline solves ambiguities between signs (BROTHER vs. SISTER), and specifies expressions (MEAT vs. HAMBURGER). It also provides information redundant to gesturing to support differentiation of similar signs.

3.2 System overview

The approach for facial feature extraction corresponds with that described in [4] to which the reader is directed for details. Figure 16 shows a schematic of the process, that can be divided into an image preprocessing stage and a subsequent feature extraction stage. Since the input image sequence covers the entire signing space, the signer's face region must be initially localized in each image. Afterwards this region is cropped and upscaled for further processing.

In order to localize areas of interest such as the eyes and mouth, a face graph is iteratively matched to the face region using a user-adapted active appearance model. Afterwards, a numerical description of the facial expression, head pose, line of sight, and lip outline is computed. For each image of the sequence, the extracted features are merged into a feature vector, which in the next step is used for classification.

3.3 Image preprocessing

In the context of facial analysis image, preprocessing aims to the robust localization of the face region which corresponds to the rectangle bounded by bottom lip and the eyebrows. With regard to processing speed, image analysis is limited to a small search mask. This mask is devised to find skin colored regions with suitable movement patterns only. The largest skin colored object is selected and subsequently limited by contiguous, non skin colored regions (Fig. 17). Additionally, the general skin color model is adapted to each individual.

For reducing influences of the environment, in particular reflections and different lighting conditions, general methods of image processing, such as *gray world color constancy*, are applied to the image sequence beforehand [34]. Furthermore, a reduction of shadow and glare effects is performed as soon as the face has been located [8].

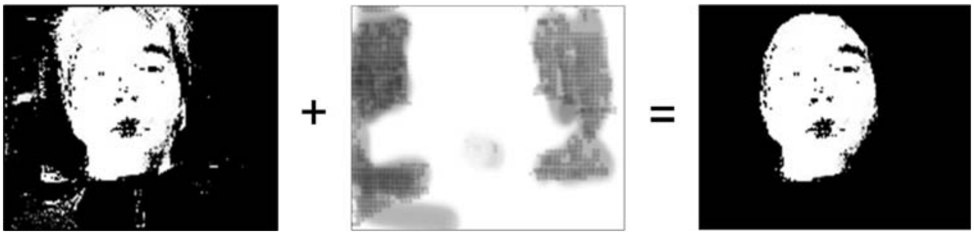


Fig. 17 Search mask composed of skin color (*left*) and motion filter (*right*)

Face localization Face localization is generally simplified by exploiting a-priori knowledge, either with respect to the whole face or to parts of it. Analytical or feature-based approaches make use of local features such as edges, intensity, color, movement, contours and symmetry, apart or in combination, to localize facial regions. Holistic approaches consider regions as a whole.

The approach described here is holistic by finding bright and dark facial regions and their geometrical relations. Eyebrows, e.g., are characterized by vertically alternating bright and dark horizontal regions. Holistic face localization makes use of three separate modules. The first module transforms the input image in an integral image for efficient calculation of features. The second module supports the automatic selection of suitable features which describe variations within a face, using Ada-Boosting. Finally, the third module is a cascade classifier that sorts out insignificant regions and analyzes in detail the remaining regions.

For side view images of faces, however, the described localization procedure yields only uncertain results. Therefore, an additional module for tracking suitable points in the facial area is applied, using the algorithm of Tomasi and Kanade [39].

3.4 Feature extraction

The interpretation of facial expression is based on so called Action Units which represent the muscular activity in a face. In order to classify these units, areas of interest, such as the eyes, eyebrows, and mouth (in particular the lips) as well as their spatial relation to each other, have to be extracted from the images. For this purpose, the face is modeled by an active appearance model (AAM), a statistical model which combines shape and texture information about human faces. Based on an eigenvalue approach the amount of data needed is reduced, hereby enabling real-time processing.

Since facial appearance is subject to high variability, the trained appearance model must be adapted to the signer. For adaptation a front view image of the signer's face is taken and applied to an artificial 3D head model. After texture matching, different synthetic views are generated in order to create a new user-specific appearance model which is then used for facial feature extraction and analysis.

Both the active appearance model approach and the adaptation of these models to a new signer is described below in more detail. With regard to the aforementioned processing chain, the localized face region is first cropped and upscaled (Fig. 18, top). Afterwards, AAMs are utilized to match the user-adapted face graph serving the extraction of facial parameters, such as lip outline, eyes, and brows.

3.4.1 *Active appearance models*

Active appearance models contain two main components: a statistical model describing the appearance of an object and an algorithm for matching this model to an example of the object in a new image [9]. In the context of facial analysis, the human face is the object and the AAM can be visualized as a face graph that is iteratively matched to a new face image (Fig. 18, bottom). The statistical models were generated by combining a model of face shape variation with a model of texture variation of a shape-normalised face. Texture denotes the pattern of intensities or colors across an image patch.

3.4.1.1 Shape model The training set consists of annotated face images where corresponding landmark points have been marked manually on each example. In this framework, the appearance models were trained on face images of 16 objects, each labelled with 70 landmark points at key positions (Fig. 19).

For statistical analysis all shapes must be aligned to the same pose, i.e., the same position, scaling, and rotation. This is performed by a Procrustes analysis which considers the shape in a training set and minimizes the sum of distances with respect to the average shape. After alignment, the shape point sets are adjusted to a common coordinate system.

For dealing with redundancy in high dimensional point sets, AAMs employ a principal component analysis (PCA). The PCA is a means for dimensionality reduction by first identifying the main axes of a cluster. Therefore, it involves a mathematical procedure that transforms a number of possibly correlated parameters into a smaller number of uncorrelated parameters, called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

With the calculated principal components it is possible to reconstruct each example of the training data. New shape instances can be approximated by deforming the mean shape $\bar{\mathbf{x}}$ using a linear combination $\mathbf{p}_s$ of the eigenvectors of the covariance matrix $\boldsymbol{\Phi}_\mathbf{s}$ as follows

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\Phi}_\mathbf{s} \cdot \mathbf{p}_s \tag{7}$$

Essentially, the points of the shape are transformed into a modal representation where modes are ordered according to the percentage of variation that they explain. By varying the elements of the shape parameters $\mathbf{p}_s$ the shape $\mathbf{x}$ may be varied as well. Figure 20 depicts the average shape and exemplary landmark variances.

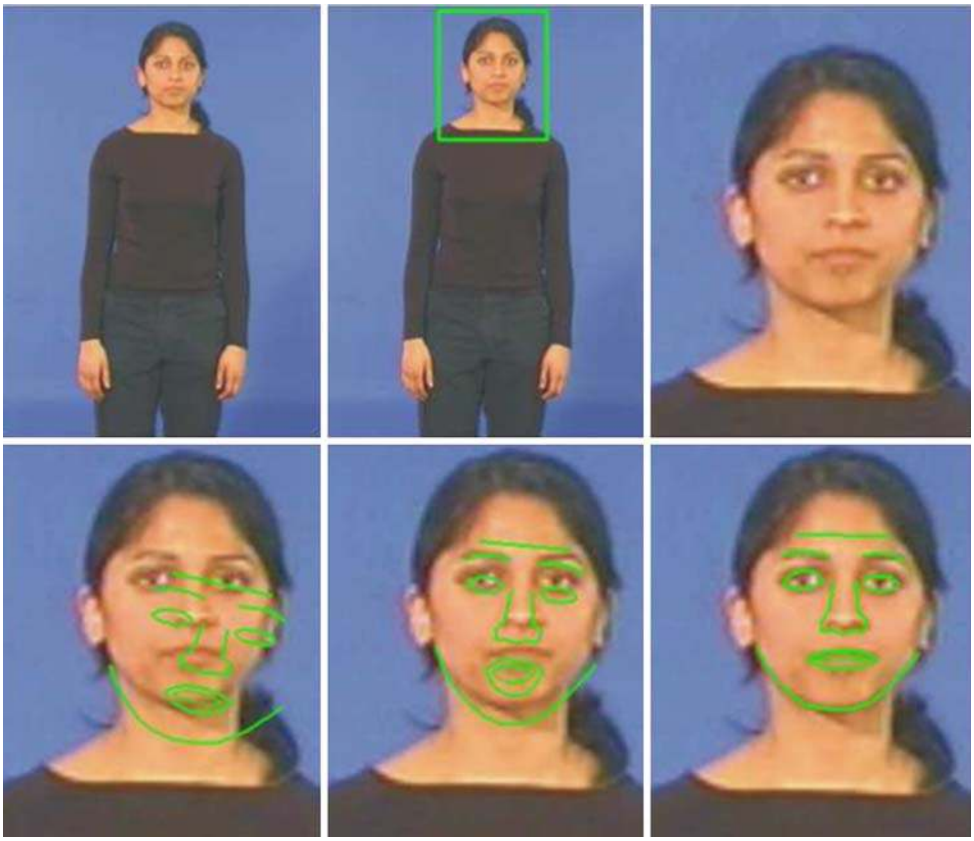


Fig. 18 Processing scheme of the face region cropping and the matching of an adaptive face graph

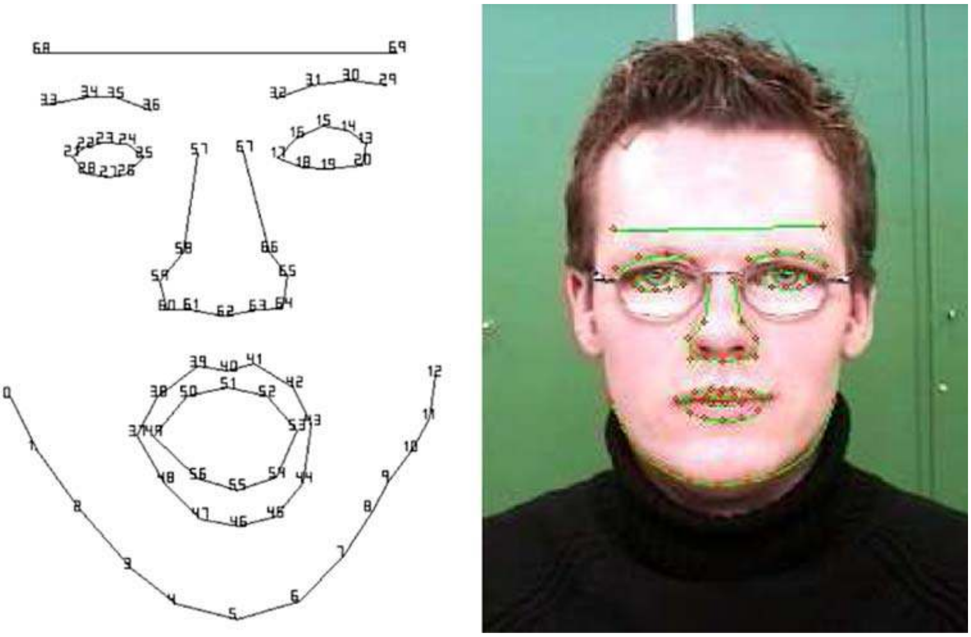


Fig. 19 Face graph with 70 landmark points (*left*) and its application to a specific user (*right*)

The eigenvalue λ_i is the variance of the ith parameter $\mathbf{p}_{\mathrm{si}}$ over all examples in the training set. Limits are set in order to make sure that a newly generated shape is similar to the training patterns. Empirically, it was found that a maximum deviation for the parameter $\mathbf{p}_{\mathrm{si}}$ should be no more than $\pm 3\sqrt{\lambda_i}$ (Fig. 21).

3.4.1.2 Texture model Data acquisition for shape models is straightforward, since the landmarks in the shape vector constitute the data itself. In the case of texture analysis, one needs a consistent method for collecting the texture information between the landmarks, i.e., an image sampling function needs to be established. Here, a piece-wise affine

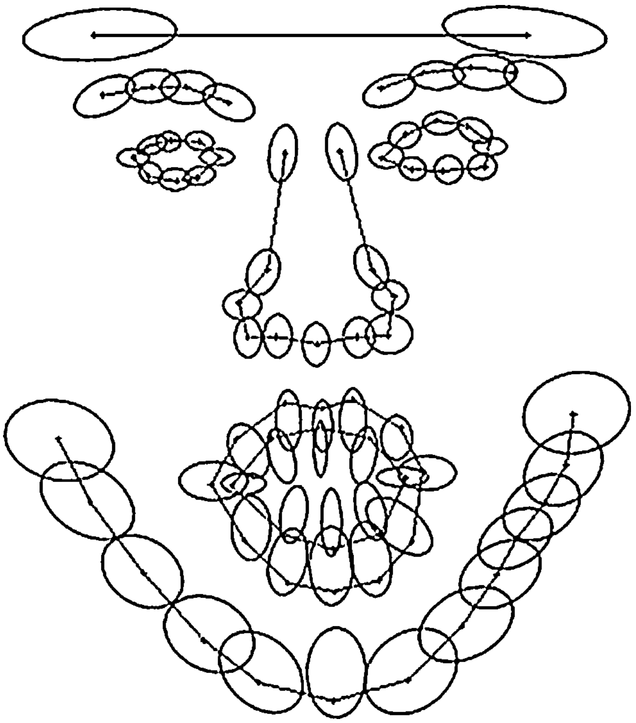


Fig. 20 Average outline and exemplary landmark variances

warp based on the Delaunay triangulation of the mean shape is applied.

Following the warp from an actual shape to the mean shape, a normalization of the texture vector set is performed to avoid the influence from global linear changes in pixel intensities. Hereafter, the analysis is identical to that of the shapes. By applying PCA, a compact representation is derived to deform the texture in a manner similar to what is observed in the training set

$$\mathbf{g} = \bar{\mathbf{g}} + \boldsymbol{\Phi}_{\mathbf{t}} \cdot \mathbf{p}_t \tag{8}$$

where $\bar{\mathbf{g}}$ is the mean texture, $\boldsymbol{\Phi}_{\mathbf{t}}$ denotes the eigenvectors of the covariance matrix and finally $\mathbf{p}_t$ is the set of texture deformation parameters.

3.4.1.3 Appearance model The appearance of any example face can thus be summarised by the shape and texture model parameters $\mathbf{p}_{\mathrm{s}}$ and $\mathbf{p}_t$. In order to remove correlation between both parameters (and to make the model representation even more compact) a further PCA is performed. The combined model obtains the form

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{Q}_{\mathbf{s}} \cdot \mathbf{c} \tag{9}$$

$$\mathbf{g} = \bar{\mathbf{g}} + \boldsymbol{Q}_{\mathbf{t}} \cdot \mathbf{c} \tag{10}$$

where $\mathbf{c}$ is a vector of appearance parameters controlling both shape and texture of the model, and $\mathbf{Q}_{\mathbf{s}}$ and $\mathbf{Q}_{\mathbf{t}}$ are matrices describing the modes of combined appearance variations in the training set. Figure 22 presents example appearance models for variations of the first five eigenvectors between $3\sqrt{\lambda}, 0, -3\sqrt{\lambda}$.

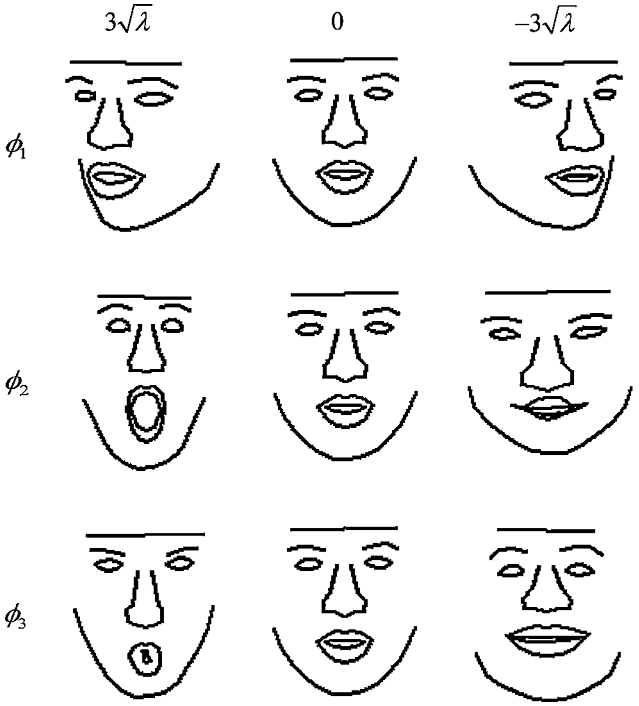


Fig. 21 Outline models for variations of the first three eigenvalues ϕ_1, ϕ_2 and ϕ_3 between $3\sqrt{\lambda}, 0$ and $-3\sqrt{\lambda}$

A face can now be synthesized for a given $\mathbf{c}$ by generating the shape-free intensity image from the vector $\mathbf{g}$ and warping it using the control points described by $\mathbf{x}$.

3.4.1.4 Active appearance model search This paragraph outlines the basic idea of AAM search. The reader interested in a detailed description is directed to [9]. In AAMs, search is treated as an optimization problem. Given a facial appearance model as described above and a reasonable starting approximation, the difference $\partial\mathbf{I}$ between the synthesized model image $\mathbf{I}_m$ and the new image $\mathbf{I}_i$ is to be minimized

$$\partial\mathbf{I} = \mathbf{I}_i - \mathbf{I}_m \tag{11}$$

By adjusting the model parameter $\mathbf{c}$ the model can deform to match the image in the best possible way.

The search algorithm exploits the locally linear relationship between model parameter displacements and the residual errors between model instance and image. This relationship can be learnt during a training phase. For this purpose, a model instance is randomly displaced from the optimum position in a set of training images. The difference between the displaced model instance and the image is recorded, and linear regression is used to estimate the

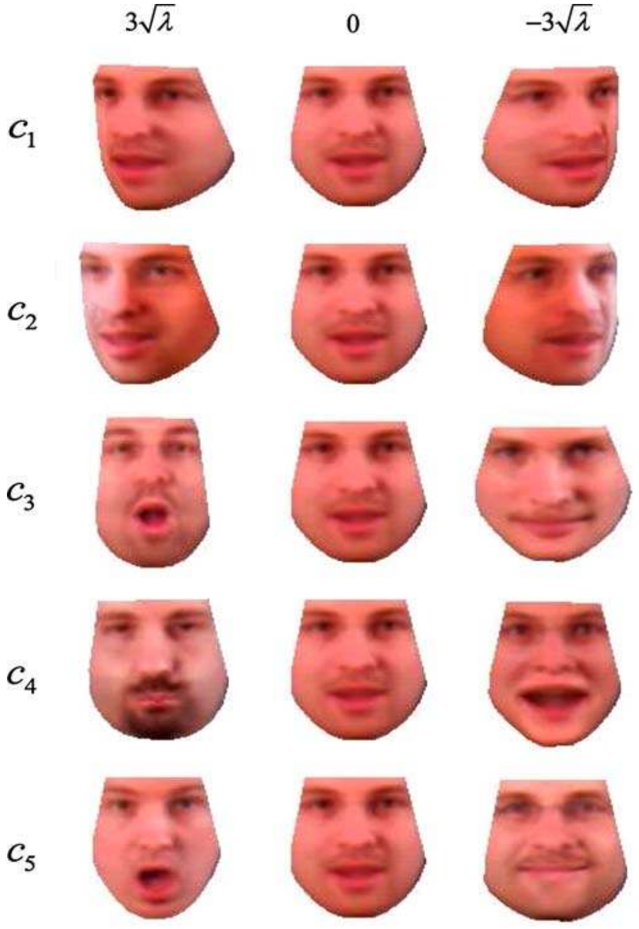


Fig. 22 Appearance for variations of the first five eigenvectors c_1, c_2, c_3, c_4 and c_5 between $3\sqrt{\lambda}, 0, -3\sqrt{\lambda}$

relationship between this residual and the parameter displacement.

During image search, the model parameters must be found that minimize the difference between image and synthesised model instance. An initial estimate of the instance is placed in the image and the current residuals are measured. The relationship is then used to predict the changes to the current parameters which would lead to a better match. A good overall match is obtained in a few iterations, even from poor starting estimates.

3.4.2 *Person-adaptive active appearance models*

Since facial appearance models are based on training sets in which the current signer is not included, it often happens that the face graph does not match accurately. In addition, special user groups, e.g., persons wearing a beard or eyeglasses, make matching difficult. For producing better results it is helpful to create a user-specific model by synthesized views. This requires a training step in which a front view image of the user is taken and applied to an artificial 3D head with an anatomic correct muscle model [7]. The muscle model also allows generating different facial expressions (Fig. 23).

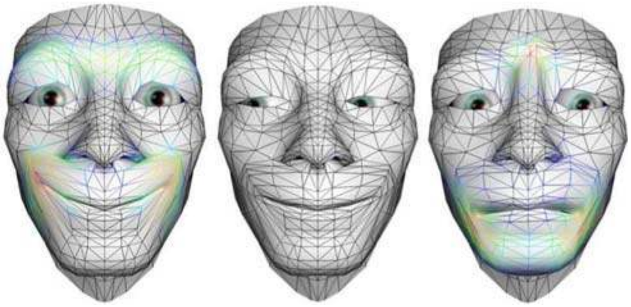


Fig. 23 The artificial model of a human head can produce different facial expressions by changing the parameters of the muscle model

In order to use the artificial head model for facial feature extraction, such model has to adapt both shape and texture information of the signer's face. In the first step of adaptation the head model is manipulated with simple transformations, such as scaling and translation. After that, inner vertices are weighted by the distance to the nearest feature vertex and moved with the weight in the x–y-layer. The z-coordinate is unchanged, because there is no information about depth by a monocular camera system. After geometric adaptation, the texture has also to be matched to the head model. If the size of the texture does not match exactly to the model, it has to be rescaled and shifted, so that the texture feature vertex has the exact same position as the head model feature vertex.

Now with the 3D head model it is possible to generate different views of the signer's face by varying head pose, facial expression, and even lighting condition. The synthetic views are then used to create a new person-adapted appearance model, which is individually adapted to the current signer.

3.5 Feature computation

After matching the face graph to the signer's face in the input image, sequence areas of interest such as his eyes, eyebrows, and mouth (in particular the lips), as well as their spatial relation to each other, can be easily extracted. Geometric features describing forms and distances serves for encoding the facial expression. These features are computed directly from the matched face graph and are divided into three groups (Fig. 24). At first, the lip outline is described by width, height and form-features like invariant moments, eccentricity and orientation. The second group contains the distances between eyes and mouth

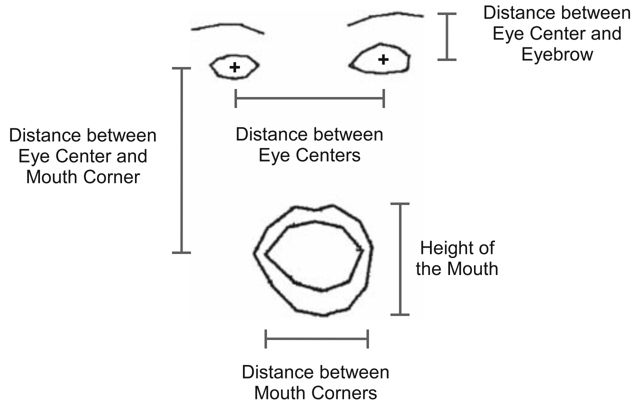


Fig. 24 Representation of facial parameters suited for sign language recognition

corners, whereas the distances between eyes and eye brows are in the third group.

More complicated is the computation of the other facial parameters: head pose, line of sight, and lip outline. These parameters cannot be extracted directly. Therefore, special algorithms were developed [7], which nevertheless rely on information derived from the face graph. These algorithms are described in the following subsections. Finally, an overlap resolution for partially overlapping of the face by the signer's hands is presented.

With regard to the processing chain for each image of the sequence, the extracted facial parameters are merged into a feature vector, which in the next step is used for classification.

3.5.1 Head pose estimation

For estimation of the head pose two approaches are pursued in parallel (Fig. 25). In the first approach, roll and pitch angle of the head are determined analytically. Calculation is done by a linear back transformation of the distorted face place on an undistorted frontal view of the face. The second, holistic approach makes use of a projection into a so-called pose eigenspace for comparing the unknown head pose with known reference poses. Finally, the results of the analytic and the holistic approach are compared. In case of significant differences, the actual head pose is estimated by utilizing the last correct result combined with a prediction involving the optical flow.

3.5.1.1 Analytic approach The analytical approach is based on the face plane, a trapezoid described by the outer corners of the eyes and mouth. These four points are taken from a matched face graph.

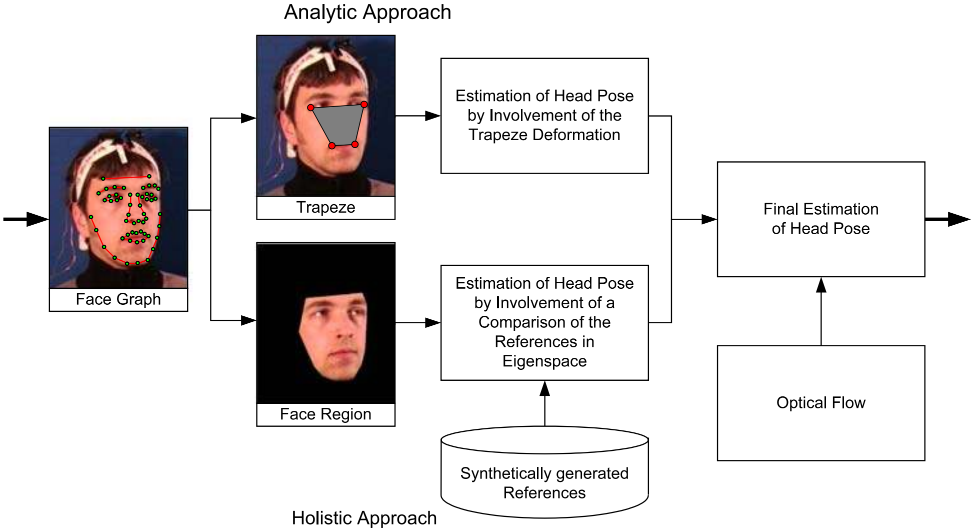
In a frontal view the face plane appears to be symmetrical. If, however, the view is not frontal, the area between eyes and mouth will be distorted. In order to calculate roll σ and pitch angle τ, the distorted face plan is transformed into a frontal view (Fig. 26). A point $\underline{x}$ on the distorted plane is transformed into an undistorted point $\underline{x}'$ by

$$\underline{x}' = U\underline{x} + \underline{t} \qquad (12)$$

where U is a linear transformation matrix and $\underline{t}$ a translation. The matrix U can be decomposed into an isotropic scaling, a scaling in the direction of τ, and a rotation around the optical axis of the virtual cam.

Roll and pitch angles are always ambiguous due to the implied trigonometric functions, i.e., the trapezoid described by mouth and eye corners is identical for different head poses. This problem can be solved by considering an additional point such as, e.g., the nose tip, to fully determine the system.

Fig. 25 Head pose is derived by a combined analytical and holistic approach


Analytic Approach
Trapeze
Estimation of Head Pose by Involvement of the Trapeze Deformation
Face Graph
Final Estimation of Head Pose
Face Region
Estimation of Head Pose by Involvement of a Comparison of the References in Eigenspace
Optical Flow
Synthetically generated References
Holistic Approach

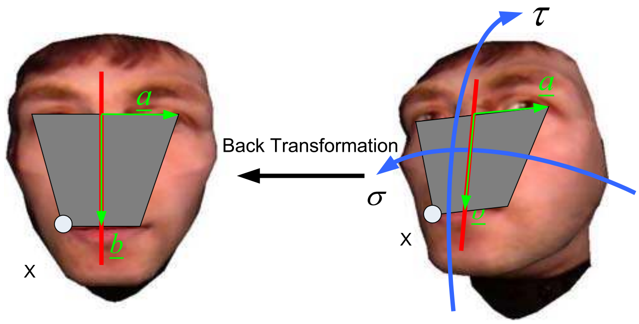


Fig. 26 Back transformation of a distorted face plane (*right*) on an undistorted frontal view of the face (*left*) which yields roll σ and pitch angle τ. Orthogonal vectors $\underline{a}$ (COG eyes—outer corner of the left eye) and $\underline{b}$ (COG eyes—COG mouth corners)

3.5.1.2 Holistic approach The holistic approach makes use of the PCA, which transforms the face region into a data space of eigenvectors that results from different reference poses. This data space is called pose eigenspace (PES). The reference poses are generated by means of a rotating virtual head and are distributed equally between − 60 and 60 degrees yaw angle. Here the face region is derived from the convex hull that contains all nodes of the face graph.

Before projection into PES, the monochrome images used for reference are first normalized by subtracting the average intensity from individual pixels and then dividing the result by the standard deviation. Variations resulting from different illuminations are averaged. After transformation the reference image sequence corresponds to a curve in the PES. This is illustrated by Fig. 27, where only the first three eigenvectors are depicted.

Now, if the pose of a new view is to be determined, the corresponding face region is projected into the PES as well, and subsequently the reference point with the smallest Euclidean distance is identified.

3.5.2 Determination of line of sight

Because the line of sight is defined by the position of both irides, they have to be localized first. For iris localization the circle Hough transformation is used, which supports the reliable detection of circular objects in an image [17]. Finally, the line of sight is determined by comparing the intensity distributions around the iris with trained reference distributions using a maximum likelihood classifier. In Fig. 28 the entire concept for line of sight analysis is illustrated.

Since the iris contains little red hue, only the red channel is extracted from the eye region which contains a high contrast between skin and iris. In this channel, a gradient-map is computed in order to emphasize the amplitude and phase between iris and its environment. The Hough transformation is applied on a search mask which is based on a threshold segmentation of the red channel image. Local maxima in the Hough space then point to circular objects in the original image. The iris by its very nature represents a homogeneous area. Hence the local extremes are being validated by verifying the filling degree of the circular area with an expected radius in the red channel.

For line of sight identification the eyes' sclera is analyzed. Following iris localization, a concentric circle is described around it with the double of its diameter. The resulting annulus is divided into eight circular arc segments. The intensity distribution of these segments then indicates the line of sight. This is illustrated by Fig. 29, where distributions for two different lines of sight are presented, which are similar for both eyes but dissimilar for different lines of sight. A particular line of sight associated with an intensity distribution is then identified with a maximum likelihood classifier. For classifier training a sample of 15 different lines of sights were collected from ten subjects.

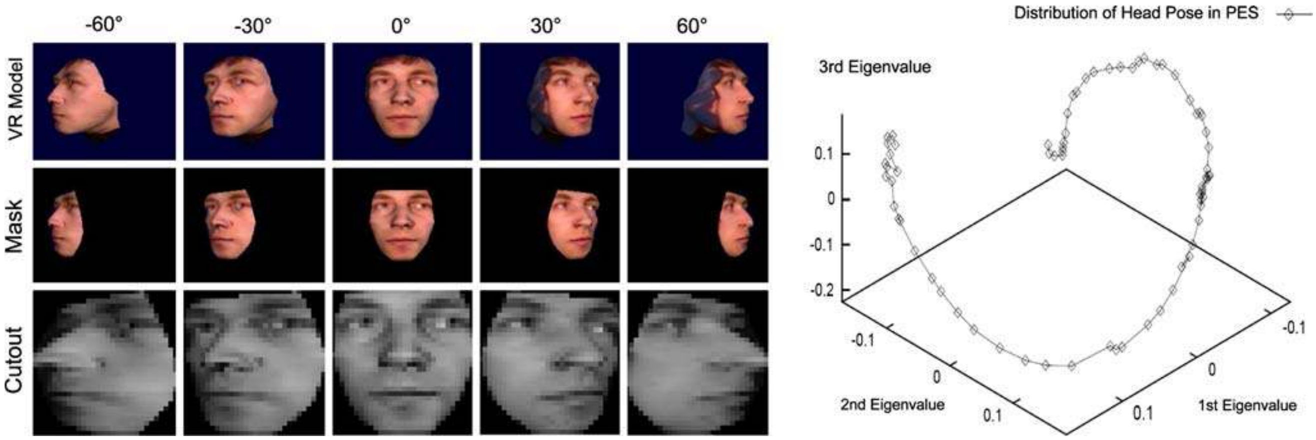


Fig. 27 Holistic approach making use of the principal component analysis. *Top* Five of 60 synthesized views. *Middle* Masked faces. *Bottom* Cropped faces. *Right* Projection into the pose eigenspace

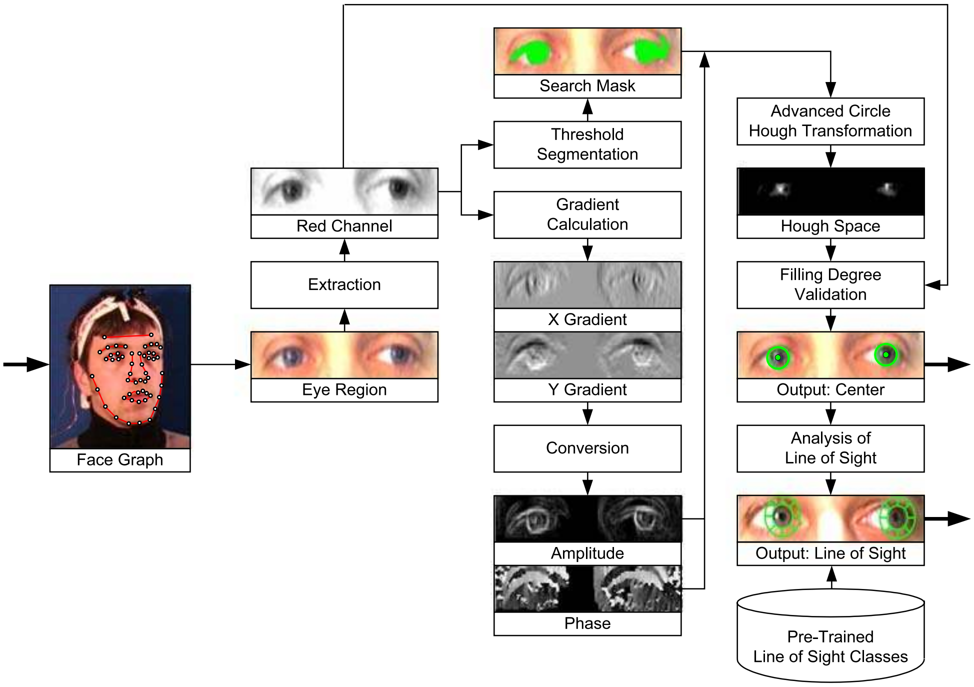


Fig. 28 Line of sight is identified based on amplitude and phase information in the red channel image. An extended Hough transformation applied to a search mask finds circular objects in the eye region

3.5.3 *Determination of lip outline*

The extraction of lip outlines is based on an active shape model (ASM), an iterative algorithm for matching a statistical model of object shape to a new image. Though related to active appearance models, ASMs do not incorporate any texture information. The statistical model is given by a point distribution model (PDM) which represents the shape and its possible deformation of the lip outline. For ASM initialization the lip borders must be segmented from the image as accurately as possible.

3.5.3.1 Lip region segmentation Segmentation of the lip region makes use of four different feature maps which all emphasize the lips from the surrounding by color and gradient information (Fig. 30).

The first two maps enhance the contrast between lips and surrounding skin by exploiting different color spaces. For this purpose, several color spaces were investigated. The results showed that the nonlinear LUX (Logarithmic hUe eXtention) color space [26] and the $I_1I_2I_3$ color space are most suited.

The third map represents the probability that a pixel belongs to the lips. The required ground truth, i.e., lip color histograms, has been derived from 800 images segmented by hand. The a posteriori probabilities for lips and background are then calculated using the Bayes theorem.

The fourth map utilizes a Sobel operator that emphasizes the edges between the lips and skin- or beard-region. This

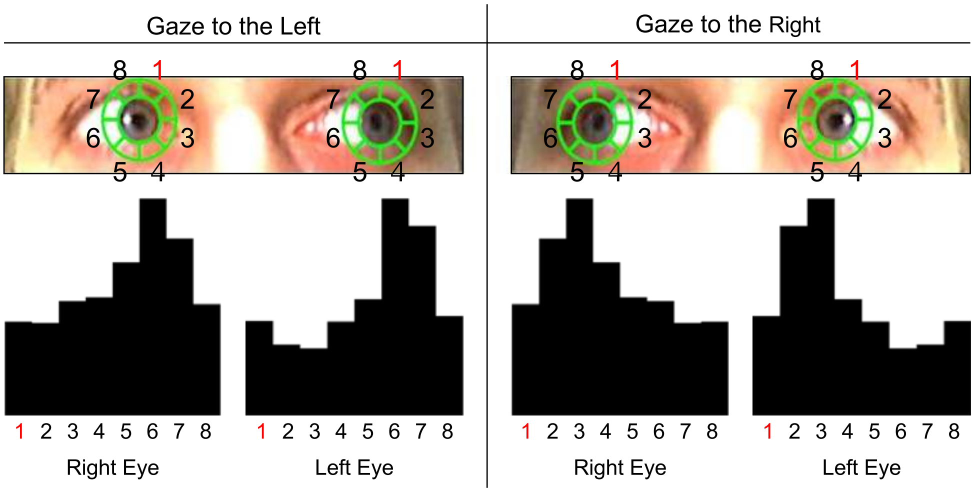


Fig. 29 Line of sight identification by analyzing intensities around the pupil

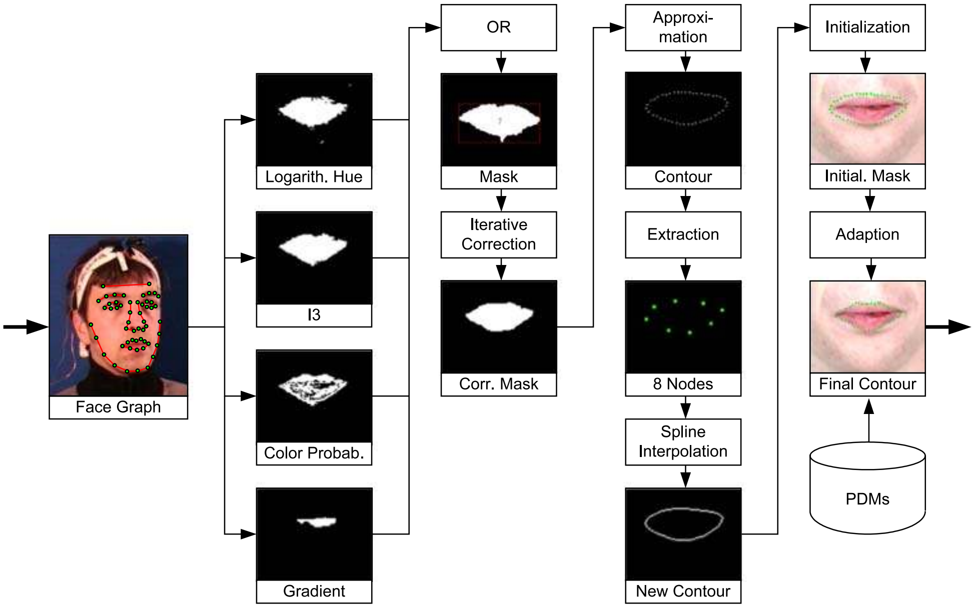


Fig. 30 Identification of lip outline with an active shape model. Four different features contribute to this process

gradient map serves the union of single regions in cases where upper and lower lips are segmented separately, due to dark corners of the mouth or to teeth. The filter mask is convoluted with the corresponding image region.

Finally, the four different feature maps need to be combined for establishing an initialization mask. For fusion, a logical OR without individual weighting was selected, as weighting did not improve the results.

3.5.3.2 Lip modeling For the collection of ground truth data, mouth images were taken from 24 subjects with the head filling the full image format. Each subject had to perform 15 visually distinguishable mouth forms under two different illumination conditions. Subsequently, upper and lower lip, mouth opening, and teeth were segmented manually in these images. Then 44 points were equally assigned on the outline with point 1 and point 23 being the mouth corners.

Since the segmented lip outlines vary in size, orientation, and position in the image, all points have to be normalized accordingly. The average form of training lip outlines, their eigenvector matrix and variance vector, form the basis for the PDMs. The resulting models are depicted in Fig. 31, where the first four eigenvectors have been varied in a range between $3\sqrt{\lambda}, 0, -3\sqrt{\lambda}$.

3.5.4 Overlap resolution

In case of partially overlapping of the face by one or both hands, an accurate fitting of the active appearance models is usually no longer possible. Furthermore, it is problematic that the face graph is often computed astounding precisely, even if there is not enough face texture visible. In this case, a determination of the features in the affected regions is no longer possible. For compensation of this effect, an additional module is involved, that evaluates all specific regions separately with regard to hands' overlappings.

In Fig. 32 two typical cases are presented. In the first case, one eye and the mouth are hidden, so the feature-vector of the facial parameters is not used for classification. In the second case, the overlapping is not critical for classification. Hence the facial features are consulted for the classification process.

The hand tracker indicates a crossing of one or both hands with the face once the skin colored surfaces touch each other. In these cases it is necessary to decide whether the hands affect the shape substantially. Therefore, an ellipse for the hand is computed by using the manual parameters. In addition, an oriented bounding box is drawn around the lip contour of the active appearance shape. If the hand ellipse touches the bounding box, the Mahalanobis distance of the shape fitting determines the decision. If this is too large, the shape is marked as invalid. Since the Mahalanobis distance of the shapes depends substantially on the trained model, not an absolute value is used here, but a proportional worsening. Experiments have shown that a good overlapping recognition can be achieved if 25% of the face is hidden.

4 Statistical classification

Having discussed the feature extraction stage in detail, this section focuses on statistical classification methods suited

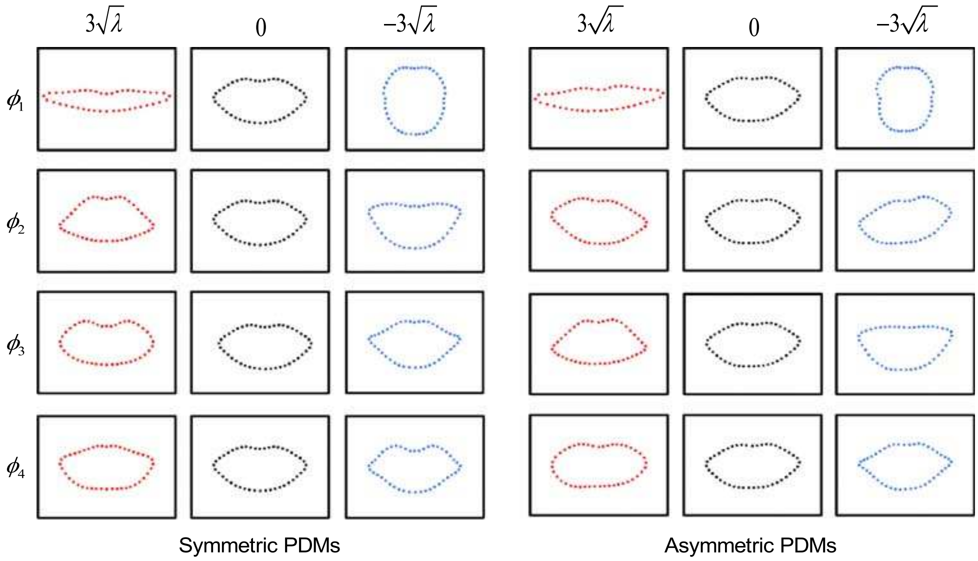


Fig. 31 Point distribution models for variations of the first four Eigenvectors ϕ_1, ϕ_2, ϕ_3, ϕ_4 between $3\sqrt{\lambda}, 0, -3\sqrt{\lambda}$

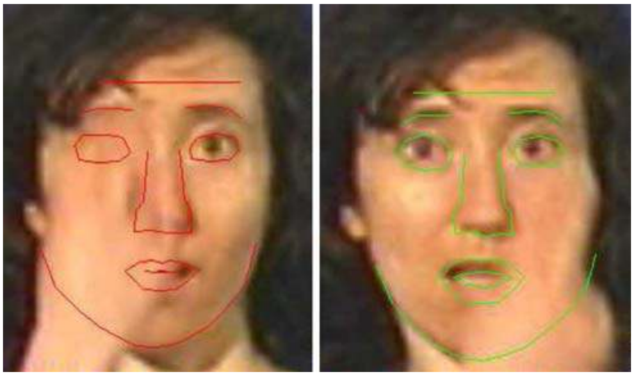


Fig. 32 During the overlapping of the hands and face several regions of the face are evaluated separately. If e.g. mouth and one eye could be hidden (*left*), no features of the face are considered. However, if eyes and mouth are located sufficiently the won features could be used for the classification (*right*)

for isolated and continuous sign language recognition. Statistical classification requires that, for each sign of the vocabulary to be recognized, a reference model must be build beforehand. Depending on the linguistic concept, a reference model represents a single sign either as a whole or as a composition of smaller subunits—similar to phonemes in spoken languages. The corresponding models are therefore called *word models* and *subunit models*, respectively.

The choice of the recognition approach generally depends on the vocabulary size and the availability of sufficient training data for creating effective reference models. While the application of recognition systems based on word models is limited to rather small vocabularies, systems based on subunit models are able to handle larger vocabularies. This limitation results from the following training problem. In order to adequately train a set of word models, each word in the vocabulary must appear several times in different contexts. For large vocabularies, this implies a prohibitively large training set. Moreover, the recognition vocabulary may contain words which had not appeared in the training phase. Consequently, some form of word models compositions technique is required to generate models for those words which have not been seen sufficiently during training.

According to the different classification concepts for sign language recognition, this section is divided into two parts: the first covers recognition based on word models for small vocabularies (Sect. 4.1), and the second deals with recognition based on subunit models for large vocabularies (Sect. 4.2).

4.1 Recognition using word models

The components of a sign language recognition system based on word models are shown in Fig. 33. For each frame of the input image sequence, the feature extraction stage creates a feature vector that reflects the manual and facial parameters. Due to the nature of sign language, the following additional processing step is advisable. Cropping leading and/or trailing frames in which both hands are idle speeds up classification and prevents the classifier from processing input data that carries no information.

The recognition system operates in two different modes. In training mode, the presented sign is known. The feature vectors received from the feature extraction stage are used to build statistical models which represent the knowledge regarding how signs were performed. Training results in a model database containing one word model for each sign in the vocabulary. When the system is switched to recognition

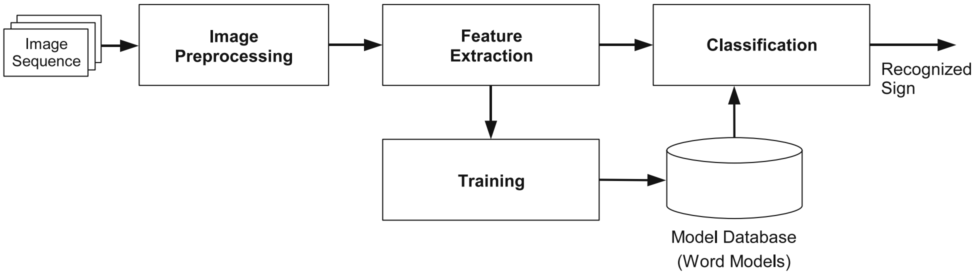


Fig. 33 Components of the training/classification process for word models

mode, the word models allow identification of an unknown sign by means of a comparison of its features.

However, similar to speech, the articulation of a sign generally varies in speed and amplitude. Even if the same person performs a same sign twice, small differences in manual configuration and facial expression will occur. Hidden Markov models (HMMs) are suited to solve these problems of sign language recognition. The ability of HMMs to compensate time and amplitude variances of signals has been proven in the context of speech and character recognition [32].

This subsection is structured as follows. At first, the basic theory of HMMs is summarized. The reader interested in a deep introduction is directed to [16, 31]. Afterwards, the classification methods for both isolated and continuous sign language recognition are described in more detail.

4.1.1 Hidden Markov models

A hidden Markov model is a finite state machine which makes a state transition once every time instant, and each time a state is entered, an observation vector is generated according to a probability density function associated with that state. Transitions between states are also modeled probabilistically describing another stochastic process. Since only the output and not the state itself is visible to an external observer, the state sequence is *hidden* to the outside. More briefly, an HMM is a doubly embedded stochastic process with an underlying stochastic process that is not observable.

Using a compact notation, an HMM λ can be completely described by its parameters $\lambda = (A, B, \Pi)$. Each parameter specifies a different probability distribution as follows. The matrix $A = \{a_{ij}\}$ represents the state transition probability distribution, where a_{ij} is the probability of taking a transition from state s_i to state s_j. The parameter $B = \{b_j(\mathbf{o_t})\}$ defines the output probability distribution, with $b_j(\mathbf{o_t})$ denoting the probability of emitting an observation vector $\mathbf{o_t}$ at time instant t when state s_j is entered. This probability is usually expressed by a continuous distribution function, which is in many cases a mixture of Gaussian distributions. Finally, the vector $\Pi = \{\pi_i\}$ defines the initial state distribution, whose elements describe the probability of starting in state s_i.

For reducing computational cost, several assumptions are commonly made in practice. The Markov assumption states that the transition probabilities are modeled as a first order Markov process, i.e., the probability of taking a transition to a new state only depends on the previous state and not on the entire state sequence. Moreover, stationarity is assumed, i.e., the transition probabilities are independent of the actual time at which the transition takes place. Another assumption, called the output-independence assumption, expresses that an observation only depends on the current state and is thus statistically independent of the previous observations.

Although many different types of HMMs exist, only some of them are suited to model signals whose characteristics change over time in a successive manner. One prominent example is the Bakis model, which is widely used in the field of speech recognition. The Bakis model has the property that it can compensate different speeds of articulation. The underlying topology allows transitions to the same state, to the next state and to the one after the next state (Fig. 34).

Given the definition of HMMs above, there are three basic problems that have to be solved [32]:

- *The evaluation problem:* Given the model $\lambda = (A,B,\Pi)$ and the observation sequence $\mathbf{O} = \mathbf{o_1},\mathbf{o_2},...,\mathbf{o_T}$, the problem is how to compute $P(\mathbf{O}|\lambda)$, the probability that this observed sequence was produced by the model.
- *The estimation problem:* This problem concerns how to adjust the model parameters $\lambda = (\Pi,A,B)$ to maximize $P(\mathbf{O}|\lambda)$ given one or more observation sequences $\mathbf{O}$. The parameters must be optimized so as to best describe how the observations have come out.
- *The decoding problem:* Given the model λ and the observation sequence $\mathbf{O}$, what is the most likely state sequence $\mathbf{q} = q_1,q_2,...,q_T$ with $q_i \in \{s_1,...,s_N\}$ according to some optimality criterion? This relates to recovery of the hidden part of the model.

The decoding problem can be solved efficiently by means of the Viterbi algorithm, a formal technique for finding the

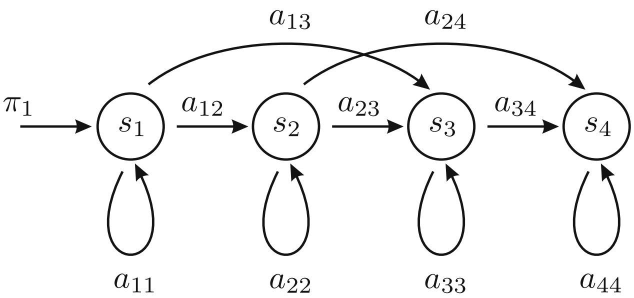


Fig. 34 Illustration of a four-state Bakis model with accompaying state transition probabilities

best state sequence. The former two problems are dealt with next when describing the training and classification module of the recognition system.

4.1.2 Classification of isolated signs

Classification requires that for each sign of the vocabulary an HMM λ_i must be build beforehand. This is performed by a prior training process. The training process is also outlined for completeness.

4.1.2.1 Training The training of hidden Markov models corresponds with the estimation problem mentioned above. There is no known way to analytically solve for the model parameter set (A, B, Π) that maximizes the probability of the observation sequence in a closed form. However, the parameter set can be chosen such that its likelihood $P(\mathbf{O}|\lambda)$ is locally maximized using an iterative procedure, such as the Baum-Welch algorithm [33].

In most practical applications a different approach, called the Viterbi training, is employed. It produces practically the same estimation, but is computationally less expensive. Given a set of observation sequences $\mathbf{O}$ the model parameters are iteratively adjusted until convergence. In each iteration, the most likely path through the associated HMM λ_i is calculated by the Viterbi algorithm. This path represents the new assignment of the observation vectors $\mathbf{o_t}$ to the states q_t. Afterwards the transition probabilities a_{ij}, the means and variances of all components of the output probability distributions $b_j(\mathbf{o_t})$ of each state s_j are reestimated. With a sufficient convergence the parameters of the HMM λ_i are available, otherwise a new iteration is requested.

The Viterbi training requires the following initialization step. Firstly, the number of states have to be determined for each HMM λ_i representing different articulations of the same sign. A fixed number of states for all HMMs λ_i is not suitable, since the training corpus usually contains signs of different lengths, e.g., very short signs and longer signs at the same time. Even the length of one sign can vary considerably. Therefore, the number of observation vectors in the shortest training sequence is chosen as the initial number of states for the HMM λ_i of the corresponding sign. After that, the system assigns the observation vectors $\mathbf{o_t}$ of each sequence $\mathbf{O}$ evenly to the states s_j and initialises the matrix A, i.e., all transitions are set equally probable.

4.1.2.2 Classification The classification problem can be viewed as follows. Given several competing HMMs $\Lambda = \{\lambda_i\}$ and an observation sequence $\mathbf{O}$, how is the model λ_i chosen which was most likely to generate that observation? Considering the case where the observation sequence is known to represent a single sign from a limited set of possible signs (the vocabulary V), the task is actually to compute

$$\hat{\lambda} = \operatorname*{argmax}_{\lambda_i \in \Lambda} P(\lambda_i|\mathbf{O}) \tag{13}$$

which is the probability of model λ_i given the observation $\mathbf{O}$. That means the model $\hat{\lambda}$ with the highest probability $P(\lambda_i|\mathbf{O})$ is chosen as recognition result. Using Bayes' rule

$$P(\lambda_i|\mathbf{O}) = \frac{P(\mathbf{O}|\lambda_i) \cdot P(\lambda_i)}{P(\mathbf{O})} \tag{14}$$

the task can be reduced to determining the likelihood $P(\mathbf{O}|\lambda_i)$ assuming that $P(\lambda_i)$ is constant, or can be computed from a language model using a priori knowledge, and that $P(\mathbf{O})$ does not affect the choice of model λ_i.

The classification problem therefore corresponds to the evaluation problem mentioned before. The likelihood is obtained by summing the joint probability over all possible state sequences $\mathbf{q}$ of length T, denoted by the set $\mathbf{Q}_T$, resulting

$$\begin{aligned} P(\mathbf{O}|\lambda_i) &= \sum_{\mathbf{q}\in\mathbf{Q}_T} P(\mathbf{O},\mathbf{q}|\lambda_i) \\ &= \sum_{\mathbf{q}\in\mathbf{Q}_T} \pi_{q_1} \cdot b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1},q_t} \cdot b_{q_t}(o_t) \end{aligned} \tag{15}$$

where T is the length of the given observation sequence $\mathbf{O}$. However, a brute force evaluation of (15) is intractable for realistic problems, as the number of possible state sequences is typically extremely high. The evaluation can be accelerated enormously using the efficient forward-backward algorithm which calculates $P(\mathbf{O}|\lambda_i)$ in an iterative manner [33].

4.1.3 Classification of continuous sign language

In the following, the training and classification process are outlined, along with necessary modifications for continuous sign language recognition. In this context, continuous sign language means that signs within a sentence are not separated by a pause. All possible sentences which are meaningful and grammatically well-formed are allowed. Furthermore, there are no constraints regarding a specific sentence structure.

4.1.3.1 Training Training HMMs on continuous sign language is very similar to training isolated signs. Hidden Markov modeling has the beneficial property that it can absorb a wide range of boundary information of models automatically for continuous sign language recognition. The training aims to the estimation of the model parameters for entire signs (not sentences), which are later used for the recognition procedure.

Since entire sentence HMMs are trained, variations caused by preceding and subsequent signs are incorporated into the model parameters. The model parameters of the single signs must be reconstructed from this data afterwards. The overall training is partitioned into the following components: the estimation of the model parameters for the complete sentence, the detection of the sign boundaries and the estimation of the model parameters for the single signs. For the training of the model parameters for both the entire sentence and single signs the Viterbi training is employed. After performing the training step on sentences, an assignment of feature vectors to single signs is clearly possible, and with that the detection of sign boundaries.

4.1.3.2 Classification In continuous sign language recognition a sign may begin or end anywhere in a given observation sequence. As the sign boundaries cannot be detected accurately, all possible beginning and end points have to be accounted for. Furthermore, the number of signs within a sentence is unknown at this time.

The former problem is illustrated in Fig. 35. Different paths exist to reach the boundary of a sign. One possible path needs the first three observation vectors $\mathbf{o_t}$ to get to the sign boundary, while within another assignment all observation vectors are used for reaching the sign boundary.

This converts the linear search, as necessary for isolated sign recognition, to a tree search. Obviously, a full search is not feasible because of its computational complexity for continuous sign recognition. Therefore a suboptimal search algorithm, called the beam search, is employed [19]. Instead of searching all paths, a threshold is used to consider only a group of likely candidates. These candidates are selected in relation to the state with the highest probability. Depending on that value and on a variable B_0 the threshold for each time step is defined. Every state with a calculated probability below this threshold is discarded from further considerations. The variable B_0 influences the recognition time. Having many likely candidates, i.e., a low threshold is used, recognition needs more time than considering less candidates. B_0 must be determined by experiments.

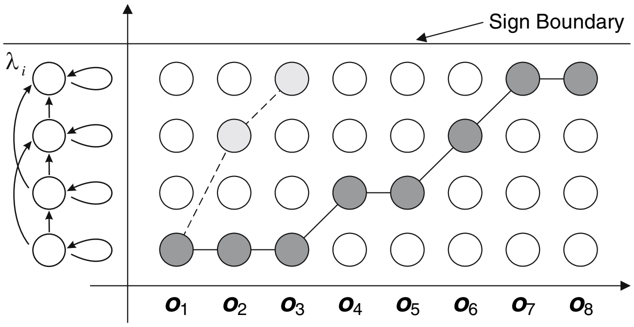


Fig. 35 Two possible paths to reach the boundary of a sign

4.2 Recognition using subunit models

As already mentioned, sign language recognition is a rather young research area compared to speech recognition. While phoneme based speech recognition systems represent today's state of the art, the early speech recognizers dealt with words as a model unity. A similar development can be observed for sign language recognition. First steps towards subunit based recognition systems have been undertaken only recently [2, 43]. This section outlines a sign language recognition system based on automatic generated subunits of signs.

4.2.1 Subunit models for signs

It is yet unclear in sign language recognition which part of a sign sentence serves as a good underlying model. Thus, most sign language recognition systems are based on word models where one sign represents one model in the model database. However, this leads to some drawbacks:

- The training complexity increases with vocabulary size.
- A future enlargement of the vocabulary is problematic as new signs are usually signed in context of other signs for training (embedded training).

Instead of modeling entire signs, it is more beneficial to model each sign as a concatenation of subunits, which is similar to modeling speech by means of phonemes. Subunits are small segments of signs, which emerge from the subdivision of signs. Figure 36 illustrates an example of the above mentioned different possibilities for model unities in the model database. The number of subunits

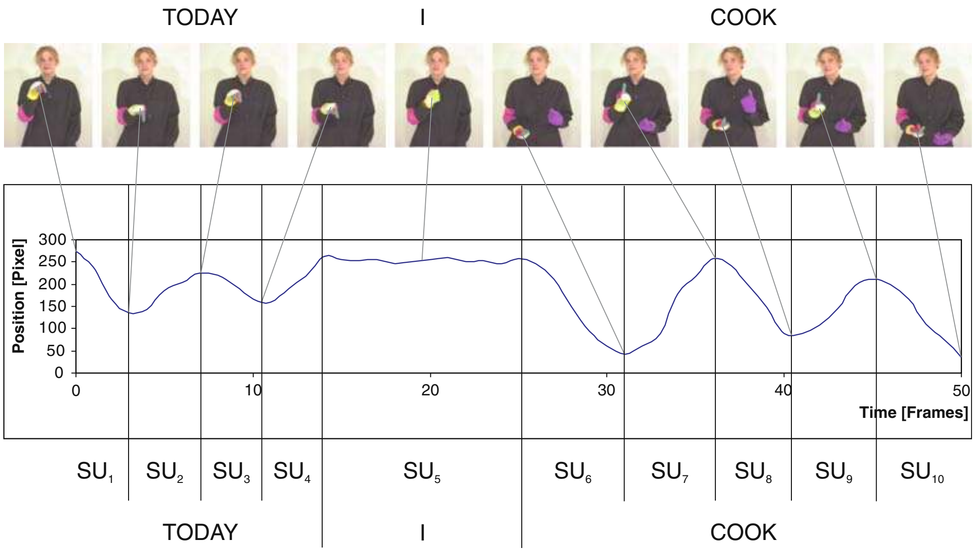


Fig. 36 The signed sentence 'TODAY I COOK' ('HEUTE ICH KOCHEN') in German Sign Language. *Top* The recorded video sequence. *Center* Vertical position of right hand during signing. *Bottom* Different possibilities to divide the sentence

should be chosen in such a way that any sign can be composed with subunits. The advantages are:

- The amount of necessary training data will be reduced, as every sign consists of a limited set of subunits.
- A further enlargement of the vocabulary is achieved by composing a new sign through concatenation of existing subunit models.
- The general vocabulary size can be enlarged.

4.2.1.1 Modifications to the recognition system A subunit based sign language recognition system needs an additional knowledge source, where the coding (also called transcription) of a sign is itemized into subunits. This knowledge source is called sign-lexicon and contains the transcriptions of the entire vocabulary. Both training and classification processes are based on this sign-lexicon. The accordant modifications to the recognition system are depicted in Fig. 37.

Modification for training The training process aims to the estimation of the subunit model parameters. The example in Fig. 37 shows that 'Sign 1' consists of the subunits (SU) SU_4, SU_7, and SU_3. The parameters of the associated hidden Markov models are trained on the recorded data of 'Sign 1' by means of the Viterbi algorithm.

Modification for classification After completion of the training process, a database is filled with all subunit models which serve as a base for the classification process. However, the aim of the classification is not the recognition of subunits, but of complete signs. Hence, again the information contained in the sign-lexicon regarding which sign consists of which subunits is needed.

4.2.2 Transcription of sign language

Subunit based recognition assumes that a sign-lexicon is available, i.e., the subunits which compose a sign are already known. This however is not the case. The subdivision of a sign into suitable subunits still poses difficult problems. In addition, the semantics of subunits have yet to be determined. The following section provides an overview of possible approaches to linguistic subunit formation.

4.2.2.1 Linguistics-orientated transcription of sign language In speech recognition subunits are mostly linguistically motivated and are typically syllables, half-syllables or phonemes. The base of this breakdown of speech is rather similar to the speech's notation system: a written text with the accordant orthography is the standard notation system for speech. Nowadays, huge speech-lexica are available consisting of the transcription of speech into subunits. These lexica are usually the base for today's speech recognizer.

When transferring this concept of linguistic breakdown to sign language recognition, one is confronted with a variety of options for notation, which all are unfortunately not yet standardized as is the case of speech. An equally accepted notation system does not exist for sign language. However, some known notation systems are examined below, especially with respect to its applicability in a recognition system. Corresponding to the term phonemes,

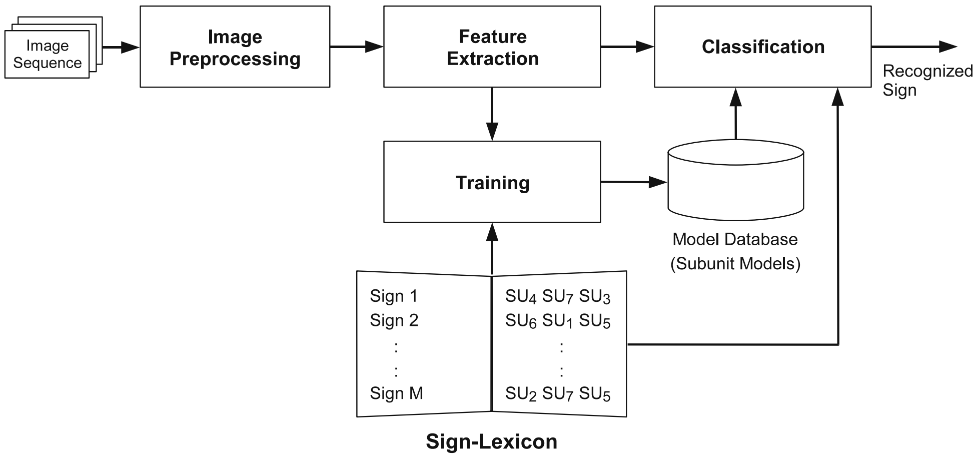


Fig. 37 Components of the recognition system based on subunits

the term *cheremes* (derived from the Greek term for ‘manual’) is used for subunits in sign languages.

Notation system by Stokoe Stokoe was one of the first to conduct research in the area of sign language linguistic in the sixties [36]. He defined three different types of cheremes. The first type describes the configuration of handshape and is called *dez* for designator. The second type is *sig* for signation and describes the kind of movement of the performed sign. The third type is the location of the performed sign and is called *tab* for tabula. Stokoe developed a lexicon for American Sign Language by means of the above mentioned types of cheremes. The lexicon consists of nearly 2500 entries, where signs are coded in altogether 55 different cheremes (12 ‘tab’, 19 ‘dez’ and 24 different ‘sig’). An example of a sign coded in the Stokoe system is depicted in Fig. 38 [36].

The employed cheremes seem to qualify as subunits for a recognition system. However, their practical employment in a recognition system turns out to be difficult. Even though Stokoe’s lexicon is still in use today and consists of many entries, not all signs are included in this lexicon. Also most of Stokoe’s cheremes are performed in parallel, whereas a recognition system expects subunits in subsequent order.

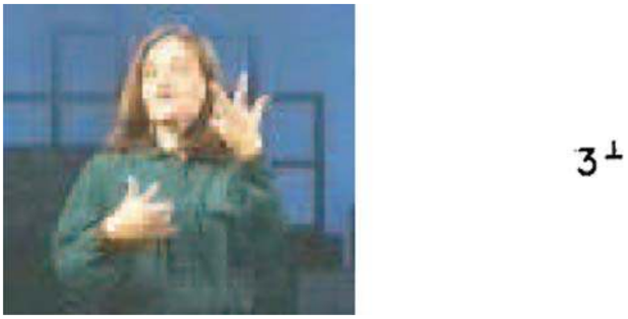


Fig. 38 Notation of the sign THREE in American Sign Language by Stokoe (from [38])

Furthermore, none of the signations cheremes (encoding the movement of a performed sign) are necessary for a recognition system, as movements are modeled by HMMs. Hence, Stokoe’s lexicon is a very good linguistic breakdown of signs into cheremes. However, without manual alterations, it is not useful as a base for a recognition system.

Notation system by Liddell and Johnson Another notation system was proposed by Liddell and Johnson [25]. They break signs into cheremes by a so called Movement-Hold model, which was introduced in 1984 and further developed since then. In this case, the signs are divided in sequential order into segments. Two different kinds of segments are possible: ‘movements’ are segments, where the configuration of a hand is still in move, whereas for ‘hold’-segments no movement takes place, i.e., the configuration of the hands is fixed. Each sign can be modeled as a sequence of movement and hold-segments. In addition, each hold-segment consists of articulatory features [3]. These describe the handshape, the position and orientation of the hand, movements of fingers, and rotation and orientation of the wrist. Figure 39 depicts an example of a notation of a sign by means of movement- and hold-segments.

Whereas Stokoe’s notation system is based on a mostly parallel breakdown of signs, in the approach by Liddell and Johnson a sequence of short segments is produced, which is better suited for a recognition systems. However, similarly to Stokoe’s notation system, no comprehensive lexicon is available where all signs are encoded. Moreover, the detailed coding of the articulatory features might cause additional problems. The video-based feature extraction of the recognition system might not be able to reach such a high level of detail. Hence, the Movement-Hold notation system is not suitable for a sign-lexicon within a recognition system without manual modifications or even manual transcription of signs.

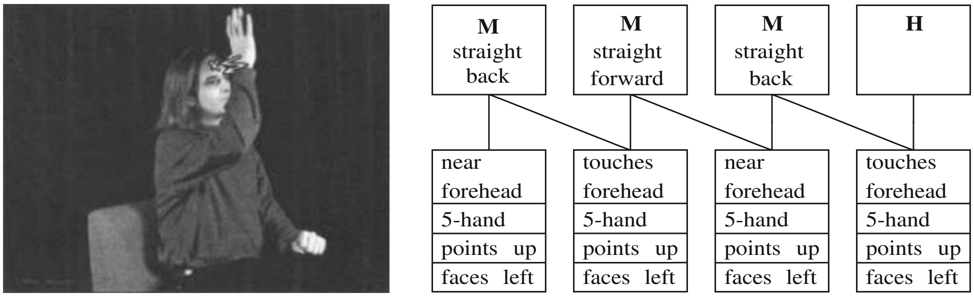


Fig. 39 Notation of the sign FATHER in American Sign Language by means of the Movement–Hold model (from [43])

4.2.2.2 Visually-orientated transcription of sign language The visual[1] approach of a notation (or transcription) system for sign language recognition does not rely on any linguistic knowledge about sign languages—unlike the two approaches described before. Here, the breakdown of signs into subunits is based on a data-driven process, i.e., no other knowledge source except the data itself is required. In a first step each sign of the vocabulary is divided sequentially into different segments, which have no semantic meaning. A subsequent process determines similarities between the identified segments. Similar segments are then pooled and labeled. They are deemed to be one subunit. Each sign can now be described as a sequence of the contained subunits, which are distinguished by their labels. This notation is also called *fenonic baseform* [19]. Figure 40 depicts as an example the temporal horizontal progression (right hand) of two different signs.

The performed signs are initially rather similar. Consequently, both signs are assigned to the same subunit (SU_3). However, the further progression differs significantly. While the gradient of 'Sign 2' is going upwards, the slope of 'Sign 1' decreases. Hence, the subsequent transcription of both signs differs.

4.2.3 Sequential and parallel breakdown of signs

The example in Fig. 40 illustrates only one aspect of the performed sign: the horizontal progression of the right hand. Regarding sign language recognition and their feature extraction, this aspect would correspond to the x-coordinate of the right hand's location. However, for a complete description of a sign, one feature is not sufficient. In fact, a recognition system must handle many more features which are merged in so called *feature groups*. The composition of these feature groups must take the linguistic sign language parameters into account, which are hand location, hand shape, and hand orientation.

Further details in this section refer to an example of separation of a feature vector into a feature group 'pos', where all features regarding the position (pos) of the two hands are grouped. Another group represents all features describing the 'size' of the visible part of the hands (size), whereas the third feature group 'dist' comprises all features regarding distances between all fingers (dist). The latter two groups stand for the sign parameter hand shape and orientation. Note that these are only examples of how to model a feature vector and its accordant feature groups. Many other ways of modeling sign language are conceivable, and the number of feature groups also may vary. To demonstrate the general approach, this example makes use of the three feature groups 'pos', 'size' and 'dist' mentioned above.

Following the parallel breakdown of a sign, each resulting feature group is segmented in sequential order into subunits. The identification of similar segments is not carried out on the entire feature vector, but only within each of the three feature groups. Similar segments finally stand for the subunits of one feature group. Pooled segments of the feature group 'pos', for example, now represent a certain location independent of any specific hand shape and orientation. The parallel and sequential breakdown of the signs finally yields three different sign lexica, which are combined to one (see also Fig. 41). Fig. 42 shows examples of similar segments of different signs according to specific feature groups.

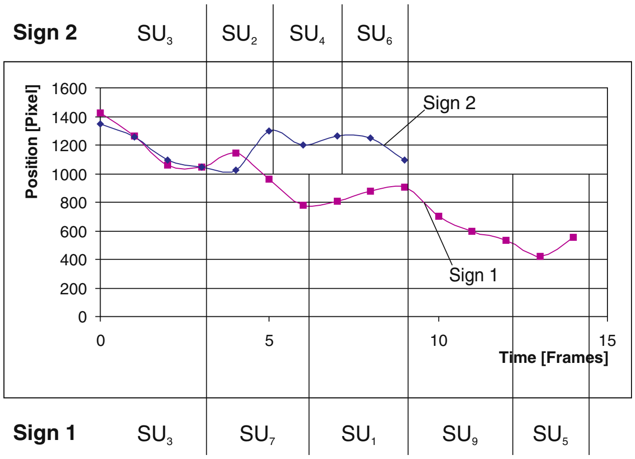


Fig. 40 Example for different transcriptions of two signs

[1] For speech-recognition the accordant name is acoustic subunits. For sign language recognitions the name is adapted.

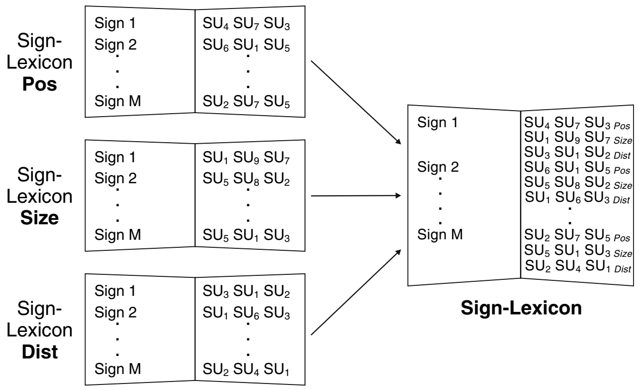


Fig. 41 Each feature group leads to an own sign-lexicon, which are finally combined to one sign-lexicon

4.2.4 Modification to parallel hidden Markov models

The breakdown of signs into feature groups means that the sequential feature vector sequence is split in several parallel signals. Since conventional HMMs are suited to handle sequential signals, a different statistical approach is required for modeling sign language. However, handling of parallel signals can be achieved by using multiple HMMs in parallel, one for each feature group. This concept is known as parallel hidden Markov models (PaHMMs) [14]. The parallel HMMs, each called a *channel*, are independent from each other, i.e., the state probabilities of one channel do not influence any of the other channels.

Figure 43 depicts an example PaHMM with three channels. The last state, called a confluent state, combines the probabilities of the different channels to one probability, valid for the entire sign. The combination of probabilities is determined by the following equation:

$$P(\mathbf{O}|\lambda) = \prod_{j=1}^{J} P(\mathbf{O}_j|\lambda_j) \qquad (16)$$

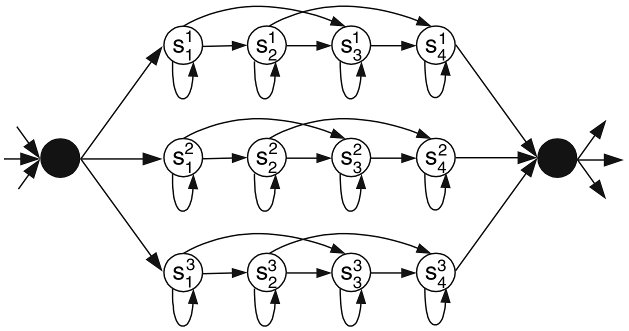


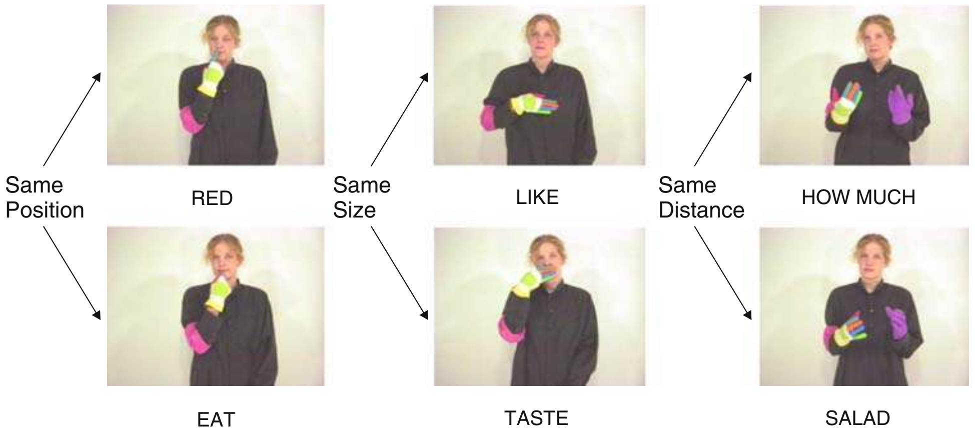
Fig. 43 Example PaHMM with three channels (Bakis topology)

The term $\mathbf{O}_j$ stands for the relevant observation sequence of one channel, which is evaluated for the accordant segment.

Modeling sign language by means of PaHMMs For recognition based on subunit models, each of the feature groups is modeled by one channel of the PaHMMs. The sequential subdivision into subunits is then conducted in each feature group separately. Figure 44 depicts the modeling of the DGS sign 'HOW MUCH' (WIEVIEL) with its three feature groups. The figure shows the *word model* of the sign, i.e., the sign and all its contained subunits in all three feature groups. Note that it is possible and highly probable that the different feature groups of a sign contain different numbers of subunit models. This is the case if, as in this example, the position changes during the execution of the sign, whereas the hand shape and orientation remains the same.

Figure 44 also illustrates the specific topology for a subunit based recognition system. As the duration of one subunit is quite short, a subunit HMM consists merely of two states. The connection of several subunits in one sign depends however on Bakis topology.

Fig. 42 Different examples for the assignment to all different feature groups



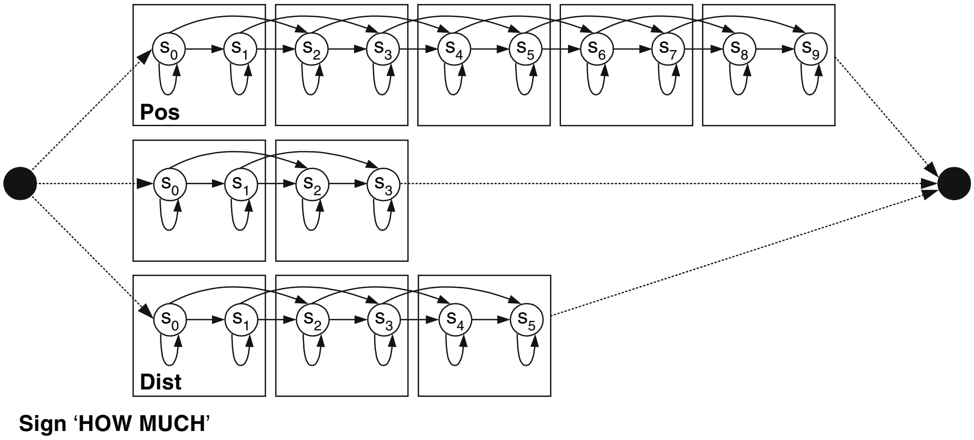


Fig. 44 Modeling the sign 'HOW MUCH' (WIEVIEL) in German Sign Language by means of PaHMMs

4.2.5 *Classification*

Determining the most likely sign sequence $\hat{W}$ which fits a given feature vector sequence **O** best results in a demanding search process. The recognition decision is carried out by jointly considering the visual and linguistic knowledge sources. Following [1], where the most likely sign sequence is approximated by the most likely state sequence, a dynamic programming search algorithm can be used to compute the probabilities $P(\mathbf{O}|W)\cdot P(W)$. Simultaneously, optimization over the unknown sign sequence is applied. Sign language recognition is then solved by matching the input feature vector sequence to all the sequences of possible state sequences, and finding the most likely sequence of signs using the visual and linguistic knowledge sources. The different steps are described in more detail in the following subsections.

4.2.5.1 Classification of isolated signs Before dealing with continuous sign language recognition based on subunit models, the general approach will be demonstrated through a simplified example of single sign recognition with subunit models. The extension to continuous sign language recognition is then described in the next section. The general approach is the same in both cases. The classification example is depicted in Fig. 45. Here, the sign consists of three subunits for feature group 'size', four subunits for 'pos' and eventually two for feature group 'dist'. It is important to note that the depicted HMM is not a random sequence of subunits in each feature group, nor is it a random parallel combination of subunits. The combination of subunits—in parallel as well as in sequence—depends on a trained sign, i.e., the sequence of subunits is transcribed in the sign-lexicon of the corresponding feature group. Furthermore, the parallel combination, i.e., the transcription in all three sign-lexica codes the same sign. Hence, the recognition process does not search *any* best sequence of subunits independently.

The signal of the example sign of Fig. 45 has a total length of 8 feature vectors. In each channel an assignment of feature vectors (the part of the feature vector of the accordant feature group) to the different states happens entirely independently from each other by time alignment (Viterbi algorithm). Only at the end of the sign, i.e., after the 8th feature vector is assigned, the so far calculated probabilities of each channel are combined. Here, the first and last states are confluent states. They are not emitting any probability, as they serve as a common beginning and end state for the three channels. The confluent end state can only be reached by the accordant end states of all channels. In the depicted example, this is the case only after feature vector $\mathbf{o_8}$, even though the end state in channel 'dist' is already reached after 7 times steps. The corresponding equation for calculating the combined probability for one model is:

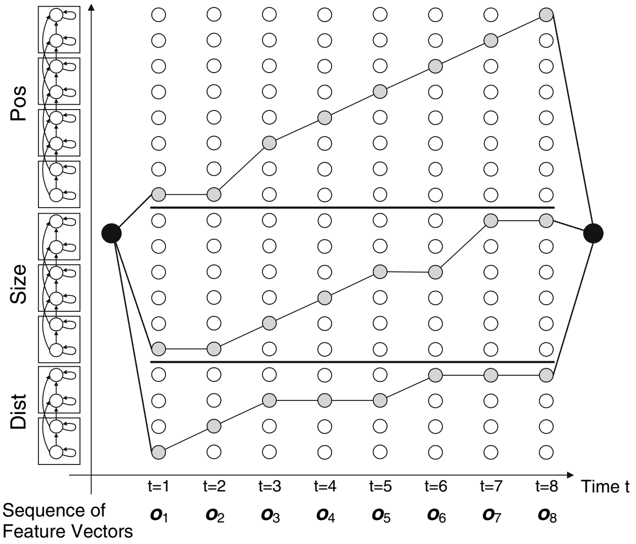


Fig. 45 Classification of a single sign by means of subunits and PaHMMs

$$P(\mathbf{O}|\lambda) = P(\mathbf{O}|\lambda_{\text{pos}}) \cdot P(\mathbf{O}|\lambda_{\text{size}}) \cdot P(\mathbf{O}|\lambda_{\text{dist}}) \tag{17}$$

The decision on the best model is reached by a maximisation over all models of the signs of the vocabulary:

$$\hat{\lambda} = \underset{\lambda_i \in \Lambda}{\operatorname{argmax}} P(\mathbf{O}|\lambda_i) \tag{18}$$

After completion of the training process, a word model λ_i exists for each sign w_i, which consists of the hidden Markov models of the accordant subunits. This word model will serve as reference for recognition.

4.2.5.2 Classification of continuous sign language In principle, the classification procedure for continuous and isolated sign language recognition is identical. However, in contrast to the recognition of isolated signs, continuous sign language recognition is concerned with a number of further difficulties, such as:

- A sign may begin or end anywhere in a given sequence of feature vectors.
- It is ambiguous how many signs are contained in each sentence.
- There is no specific order of given signs.
- Transitions between subsequent signs must be detected automatically.

All these difficulties are strongly linked to the main problem, the detection of sign boundaries. Since these can not be detected accurately, all possible beginning and end points have to be accounted for.

As introduced in the last section, the first and last state of a word model is a confluent common state for all three channels. Starting from the first state, the feature vector is divided into feature groups for the different channels of the PaHMM. From the last joint confluent state of a model a transition exists to the first confluent states of all other models. This scheme is depicted in Fig. 46.

Detection of sign boundaries At the time of classification, the number of signs in the sentence, as well as the transitions between these signs, are unknown. In order to find the correct sign transitions all models of signs are combined, as depicted in Fig. 46. The generated model constitutes one comprehensive HMM. Inside a sign model there are still three transitions (Bakis-Topology) between states. The last confluent state of a sign model has transitions to all other sign models. The Viterbi algorithm is employed to determine the best state sequence of this three-channel PaHMM. The assignment of feature vectors to different sign models becomes obvious and with it the detection of sign boundaries.

4.3 Stochastic language modeling

The classification of sign language usually depends on two knowledge sources: the visual model and the language model. Visual modeling is carried out by using HMMs as described above. Language modeling is discussed in this section. Without any language model technology, the transition probabilities between two successive signs are equal. Knowledge about a specific order of the signs in the training corpus is not utilised during recognition.

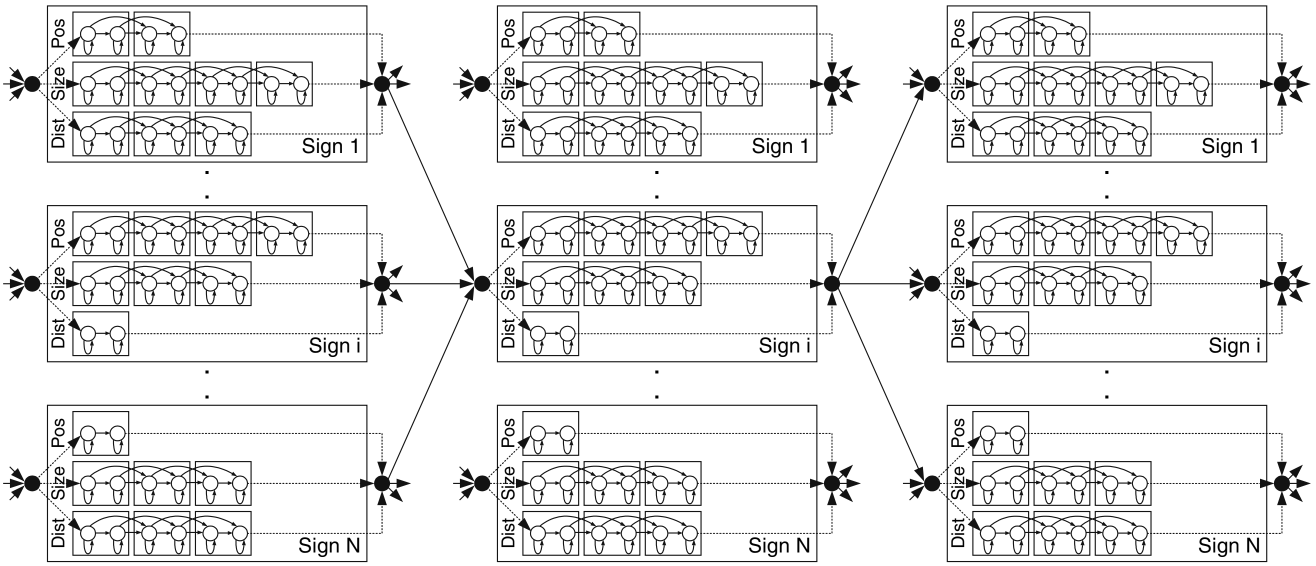


Fig. 46 A network of PaHMMs for continuous sign language recognition with subunits

In contrast, a *statistical language model* takes advantage of the knowledge that pairs of signs, i.e., two successive signs, occur more often than others. The following equation gives the probability of so called *bigram models*:

$$P(\mathbf{w}) = P(w_1) \cdot \prod_{i=2}^{m} P(w_i|w_{i-1}) \tag{19}$$

The equation estimates the probability that a given sequence of m successive signs w_i occurs. During the classification process, the probability of a subsequent sign changes, depending on the classification result of the preceding sign. By this method, typical sign pairs receive a higher probability. The estimation of these probabilities requires however a huge training corpus. Unfortunately, since training corpora do not exist for sign language, a simple but efficient enhancement of the usual statistical language model is introduced next.

Enhancement for sign language recognition The approach of an enhanced statistical language model for sign language recognition is based on the idea of dividing all signs of the vocabulary into different sign groups (SG) [2]. The probabilities of occurrence are calculated between these sign groups and not between specific signs. If a combination of different sign groups is not seen in the training corpus, this is a hint that signs of these specific sign groups do not follow each other. This approach does not require that all combinations of signs occur in the data base.

If the sequence of two signs of two different sign groups SG_i and SG_j is observed in the training corpus, *any* sign of sign group SG_i followed by *any* other sign of sign group SG_j is allowed for recognition. For instance, if the sign sequence ‘I EAT’ is contained in the training corpus, the probability that a sign of group ‘verb’ (SG_{verb}) occurs when a sign of sign group ‘personal pronoun’ ($SG_{\text{perspronoun}}$) was already seen, is increased. Therefore, the occurrence of the signs ‘YOU DRINK’ receives a high probability even though this sequence does not occur in the training corpus. On the other hand, if the training corpus does not contain a sample of two succeeding ‘personal pronoun’ signs (e.g., ‘YOU WE’), it is a hint that this sequence is not possible in sign language. As a consequence, the recognition of these two succeeding signs is excluded from the recognition process.

By this modification, a good compromise between statistical and linguistic language modeling is achieved. The assignment to specific sign groups is mainly motivated by word categories known from speech grammar. Sign groups are ‘nouns’, ‘personal pronouns’, ‘verbs’, ‘adjectives’, ‘adverbs’, ‘conjunctions’, ‘modal verbs’, ‘prepositions’ and two additional groups, which take the specific characteristics of sign languages into account.

5 Signer adaptation

Current sign language recognition systems face the problem that they achieve excellent performance for signer-dependent operation, but their recognition rates decrease significantly if the signer’s articulation deviates from the training data.

The performance drop in the case of signer-independent recognition results from the broad interpersonal variability in production of sign languages. Even within the same dialect, considerable variations are commonly present. Figure 47 shows different articulations of an example sign in British Sign Language. Analysis of the hand motion reveals that the variation between different signers is much higher than within one signer. Other manual features, such as hand shape, posture, and location, exhibit analogue variability.

As the problem of interpersonal variance cannot be solved by simple feature normalization, it must be addressed at the classification level. The most obvious solution is to increase the number of training signers. However, the recording of training data is very time-consuming, in particular for large vocabularies. Furthermore, increasing the training population usually results in lower recognition performance compared to signer-dependent systems. Hidden Markov models tend to become less accurate when covering more and more different articulations of the same sign.

Better results can be achieved with dedicated adaptation methods known from speech recognition. Such methods allow enhancing the recognition performance back to the level of signer-dependent systems. This section outlines how the sign language recognition system presented in this paper can be extended for rapid adaptation to unknown signers. A combination of maximum likelihood linear regression (MLLR) and maximum a posteriori (MAP) estimation is introduced, along with necessary modifications for signer adaptation.

5.1 System overview

Selected adaptation methods known from speech recognition are modified for the use in sign language recognition tasks to improve the performance of the signer-independent recognizer. Figure 48 shows a schematic representation of the adaptive recognition system described in this section [44].

Initially, a set of adaptation data consisting of isolated signs is collected from the unknown signer, either supervised with known transcription or unsupervised. In the latter case, the signer-independent recognizer estimates a transcription, using a confidence measure to assess the

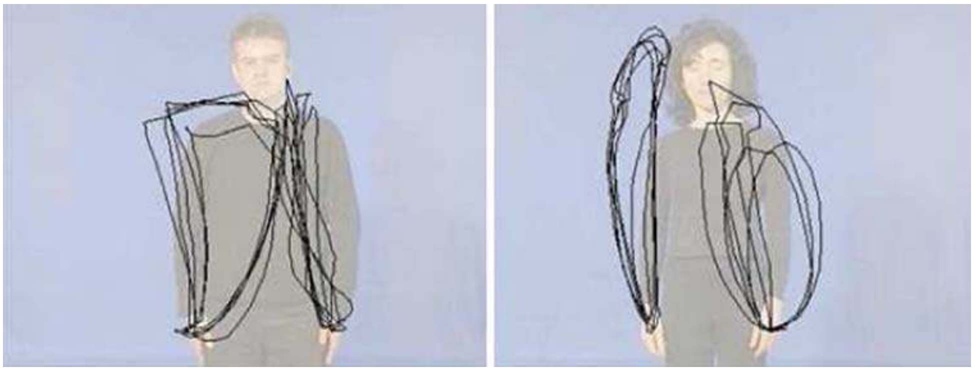


Fig. 47 The sign TENNIS in British Sign Language performed five times by two different native signers using the same dialect. Position of the hands are visualized as motion traces for comparison

quality of the recognition result. Based on the adaptation data, the adaptation process then reduces the mismatch between signer-independent models and observations from the unknown signer.

5.2 Choice of adaptation methods

Various adaptation methods have already been investigated in the context of speech recognition. Due to the obvious similarities between speech and sign language recognition, some are applicable to signer adaptation. There are generally two different adaptation approaches. While *feature-based* methods, such as vocal tract length normalization, require knowledge from the speech production domain, *model-based* approaches are well suited for adapting the recognition system.

Model-based adaptation alters the parameters of the underlying HMMs based on the given adaptation data. In the following two methods are evaluated: *maximum likelihood linear regression* and *maximum a posteriori* estimation. Both are employed in current speech recognition systems and have proven to perform excellently in the speech domain. These two approaches are introduced and modified to consider the specifics of sign languages, such as one-handed signs.

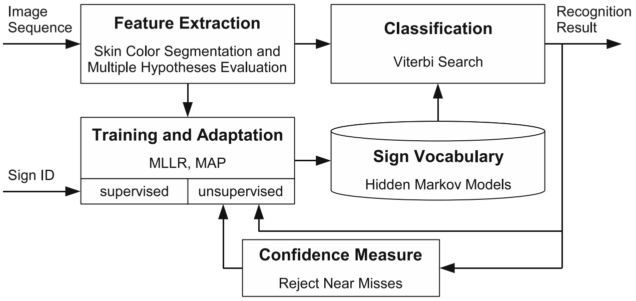


Fig. 48 Schematic of the adaptive sign language recognition system

5.3 Maximum likelihood linear regression

The mixture components of the signer-independent HMMs are clustered into a set of regression classes $C = 1,...,R$ such that each Gaussian component m belongs to one class $c \in C$. A linear transformation W_c for each class c is then estimated from the adaptation data. Estimation of the transformation matrices follows the maximum likelihood paradigm, so the transformed models best explain the adaptation sequences. Reestimation formulae for W_c based on the iterative expectation maximization algorithm are presented in [13].

The Gaussian mean μ_{m} of each component m from class c is then transformed with the corresponding matrix W_c, yielding the adapted parameter

$$\tilde{\mu}_{\mathrm{m}} = W_c \cdot \bar{\mu}_{\mathrm{m}} \tag{20}$$

where $\bar{\mu}_{\mathrm{m}}$ is the extended mean vector

$$\bar{\mu}_{\mathrm{m}}^{\mathrm{T}} = [1\ \mu_{\mathrm{m}}^{\mathrm{T}}] \tag{21}$$

A component from a model which has not been observed in adaptation data can thus be transformed based on the observed components from the same class.

As proposed in [13], a *regression class tree* is used to improve the clustering of the mixture components, where the number of regression classes depends on the available amount of adaptation data. Each node c of the tree corresponds to a regression class and a transformation W_c is associated to the node. The root contains all mixture components, yielding a global transformation W. The sons of a node form a partition of the father class, so deeper nodes yield more specialized transformations derived from fewer components. As more adaptation sequences become available, deeper transformations can be robustly estimated.

This approach is adapted to sign language recognition using explicit handling of signs that are only performed with one hand and a method for transforming models that have not been observed in the adaptation data.

5.3.1 *One-hand transformations*

The corpus contains several signs where only the dominant hand is active during the entire sequence. It is presumed that the right hand is always dominant, as features from left-handed signers are mirrored. Thus, feature extraction yields a feature vector sequence $[x_1,...,x_T]$, where for single-handed signs the entries of the non-dominant hand of each feature vector $x_t \in \mathbb{R}^{D+D}$ equal zero:

$$x_t = [0\ldots0\ x_{t,1}\ldots x_{t,D}] \tag{22}$$

Here, $x_{t,d}$ is the dth feature of the dominant hand. If HMMs are trained with such sequences, the mean vectors of the resulting mixture components have the same special form. As the adapted models should be of the same form, dedicated *one-hand transformations* are introduced.

Each class of the regression class tree containing only one-hand mixture components is marked as a one-hand class. The sons of such a class again represent one-hand classes as they form a partition of the father node. Thus each one-hand class defines a *one-hand subtree* containing only one-hand classes.

A sample regression class tree is shown in Fig. 49. The root node contains all components, represented by their mean vectors. These are either collected from one-hand or two-hand models. If a created node contains only one-hand means during tree construction, the whole subtree defined by that node will contain only one-hand classes. Such one-hand subtrees can make up a large part of the whole regression class tree.

The first half of a Gaussian parameter corresponding to a one-hand mixture component contains only zero entries, and is therefore ignored during the adaptation process. Transformations for classes that are part of a one-hand subtree are estimated from the one-hand versions of the corresponding Gaussian parameters, consisting only of the second half of mean and variance.

The use of one-hand transformations guarantees that the features for the passive hand remain passive after the transformation. Complexity of the estimation process is halved in the one-hand case due to the dimensionality reduction.

5.3.2 *Handling of unseen signs*

Sign models are called *seen* or *unseen*, depending on whether they are observed in adaptation data or not. The mixture components of an unseen HMM are transformed based on the seen components of the regression class they belong to. Although this works for large and general regression classes near the root of the tree, specialized transformations for small classes towards the tree leaves tend to produce unsatisfying results. Since the transformations are highly optimized for the seen components, the unseen components are not adapted well.

Reducing the tree size would result in broader regression classes at the tree leaves and the most special possible transformations would still be applied to a large amount of mixture components. If these general transformations are used even if more adaptation sequences become available, the effect of MLLR saturates after a certain amount of data. Thus, a special handling of the unseen components is proposed.

Not updating the unseen components at all degrades the quality of the adapted models in terms of recognition accuracy. After the transformation, the mean parameters of seen components are much closer to the range of the observations from the unknown signer than the parameters of unseen components. Thus, the Viterbi score of a model corresponding to a seen sign is likely to be higher than the score of an unseen model, so the recognizer prefers seen models in general.

This can be solved by using general transformations only for unseen components. The seen components are adapted using the most special transformation that can be robustly estimated using the regression class tree, while unseen components are adapted using a global transformation estimated at the root node of the tree.

5.4 Maximum a posteriori estimation

The maximum a posteriori estimate $\tilde{\mu}_{\mathrm{MAP}}$ for the Gaussian mean μ_{m} of a mixture component m is a linear interpolation between a-priori knowledge derived from the signer-independent model and the observations from the

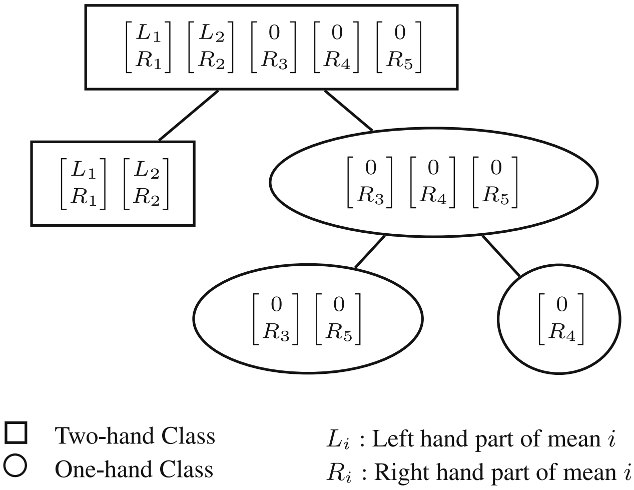

$\begin{bmatrix}L_1\\R_1\end{bmatrix}\begin{bmatrix}L_2\\R_2\end{bmatrix}\begin{bmatrix}0\\R_3\end{bmatrix}\begin{bmatrix}0\\R_4\end{bmatrix}\begin{bmatrix}0\\R_5\end{bmatrix}$
$\begin{bmatrix}L_1\\R_1\end{bmatrix}\begin{bmatrix}L_2\\R_2\end{bmatrix}$
$\begin{bmatrix}0\\R_3\end{bmatrix}\begin{bmatrix}0\\R_4\end{bmatrix}\begin{bmatrix}0\\R_5\end{bmatrix}$
$\begin{bmatrix}0\\R_3\end{bmatrix}\begin{bmatrix}0\\R_5\end{bmatrix}$
$\begin{bmatrix}0\\R_4\end{bmatrix}$
□ Two-hand Class
○ One-hand Class
L_i : Left hand part of mean i
R_i : Right hand part of mean i

Fig. 49 One-hand classes as part of the regression class tree

adaptation sequences. During Viterbi alignment of an adaptation sequence with its corresponding model, the feature vectors mapped to a certain component can be recorded, yielding the empirical mean $\bar{x}_\mathrm{m}$ of the mapped vectors. According to [22], the MAP estimate is

$$\tilde{\mu}_\mathrm{MAP} = \frac{\tau}{\tau + N} \cdot \mu_\mathrm{m} + \left(1 - \frac{\tau}{\tau + N}\right) \cdot \bar{x}_\mathrm{m} \tag{23}$$

where N is the number of feature vectors aligned to component m, and τ is a weight for the influence of the a-priori knowledge. If N approaches infinity, the influence of the signer-independent model approaches zero and the adapted parameter equals the empirical mean. Thus MAP performs well on large sets of adaptation data, but its pure form can only be used to update seen components. This can be solved by using the MLLR-adapted model as prior knowledge, replacing the signer-independent mean by the already transformed mean.

6 Performance evaluation

This section provides some performance data that has been achieved with the visual sign language recognition system presented in this paper. Performance evaluation is concerned with recognition based on word models and subunit models. In both cases recognition performance was evaluated for both isolated signs and continuous sign language.

Recognition performance for continuous sign language is typically described in terms of sign accuracy, SA, defined as:

$$SA = 1 - \frac{N_S + N_D + N_I}{N_A} \tag{24}$$

where N_A is the total number of signs in the test set, and N_S, N_D, and N_I are the total number of substitutions, deletions, and insertions respectively.

6.1 Training and test corpora

For evaluating the proposed sign language recognition system, numerous videos containing either isolated signs or continuous sign sentences were recorded and stored in two databases. According to the underlying sign language, the databases are referred to as BSL-Corpus and DGS-Corpus in the following. Each database is divided into two independent subsets, called the training and test set. While the training set is used for training the recognition system, the test set serves for performance evaluation.

In order to facilitate feature extraction, recordings were conducted under laboratory conditions, i.e., in a controlled environment with diffuse lighting and a unicolored background. The signers wear dark clothes with long sleeves and perform from a standing position (Fig. 50). Moreover, each signer was instructed to move her/his hands from a resting position beside the hips to the signing location and after signing back to the same resting position. The hands are visible throughout the whole sequence, and their start and end positions are constant and identical, which simplifies tracking.

All video clips were initially recorded on video tape and then transferred to hard disk. Image resolution is 384 × 288 pixels at 25 fps. For quick random access to individual frames, each clip is stored as a sequence of images.

6.1.1 BSL-Corpus

The BSL-Corpus was primarily built to evaluate the signer-independent recognition performance for isolated signs. For this purpose, a vocabulary of about 263 signs in British Sign Language has been recorded. The corpus consists of a base vocabulary of 153 signs and about 110 additional signs representing variations and dialects of this base vocabulary. The vocabulary comprises news items and

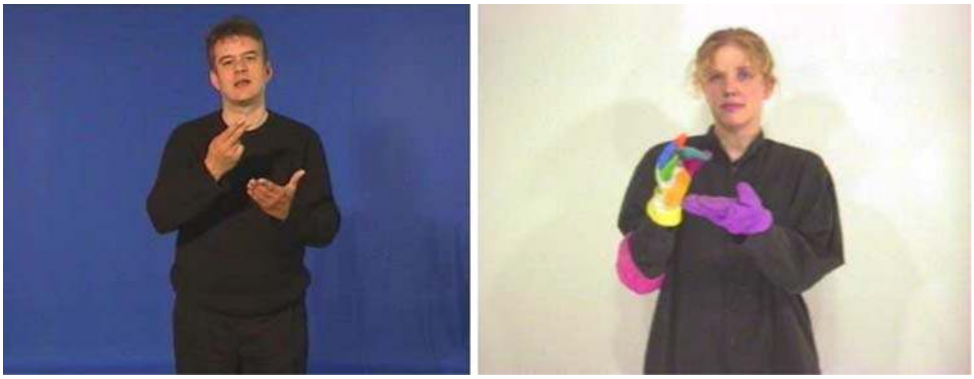


Fig. 50 Example frames taken from the BSL-Corpus (*left*) and from the DGS-Corpus (*right*), respectively

navigation commands and was not selected for discernability. As required for signer-independent recognition, most signs were performed by different native signers. While the base vocabulary was performed by 4 signers, the additional signs were articulated only by a subset of these signers. For both signer-dependent and -independent recognition, multiple productions (5–10) of each sign were recorded in order to capture typical variance and characteristic properties. The total number of video clips, each showing an isolated sign, is about 8100.

6.1.2 DGS-Corpus

The DGS-Corpus was built with the objective of evaluating the signer-dependent recognition performance for isolated and continuous sign language. The vocabulary comprises 152 signs in German Sign Language representing seven different word types such as nouns, verbs, adjectives, etc. The signs were chosen from the domain ‘shopping in a supermarket’. The entire corpus was performed by one person. The native language of the signing person is German, but she is working as an interpreter for DGS and therefore did not learn the signs explicitly for this task.

The corpus consists of a large number of videos showing each sign of the vocabulary as a single isolated sign, as well as in context of continuous signing. Based on the vocabulary, overall 631 different continuous sentences were constructed and recorded. Each sentence ranges from two to nine signs in length. No intentional pauses are placed between signs within a sentence, but the sentences themselves are separated. There are no constraints regarding a specific sentence structure. All sentences of the sign database are meaningful and grammatically well-formed. For modeling variance in articulation, each isolated sign and sentence was performed 10 times.

Training set preparation focused on the construction of sign sentences with a great number of different transitions between the signs. However, these transitions are still different from those seen in the independent test set. In order to evaluate the recognition performance for different vocabulary sizes, the corpus is divided into three subcorpora simulating a vocabulary of 52, 97, and 152 signs respectively.

6.2 Recognition using word models

This section reports some performance data for sign language recognition based on word models. Results were obtained for isolated signs and continuous sign language.

6.2.1 Classification of isolated signs

Based on the BSL-Corpus, recognition performance for isolated signs was evaluated for both signer-dependent and signer-independent operation. In the latter case, recognition rates are given for single signs under controlled laboratory condition, as well as under real world condition. Unless otherwise stated, only manual features were used for classification.

6.2.1.1 Signer-dependent recognition Table 2 presents the signer-dependent recognition performance from a leaving-one-out test for four signers and various video resolutions under controlled laboratory conditions. The training resolution was always 384 × 288. The vocabulary size is specified separately for each signer as the number of recorded signs varies slightly. Interestingly, hand coordinates alone accounted for approximately 95% recognition of a vocabulary of around 230 signs. On a 2 GHz PC, processing took an average of 11.79s/4.15s/3.08s/2.92s per sign, depending on resolution. Low resolutions caused only a slight decrease in recognition rate, but reduced processing time considerably. So far a comparably high performance has only been reported for intrusive systems.

Under the same conditions head pose, eyebrow position, and lip outline were employed as non-manual features. The recognition rates achieved on the vocabularies presented in Table 2 varied between 49.3 and 72.4% among the four signers, with an average of 63.6%. Hence, roughly two of

Table 2 Signer-dependent isolated sign recognition with manual features in controlled environments

Test video resolution	Features	Signer, vocabulary size				
		Ben, 235 signs (%)	Michael, 232 signs (%)	Paula, 219 signs (%)	Sanchu, 230 signs (%)	∅ 229 signs (%)
384 × 288	All	98.7	99.3	98.5	99.1	98.9
192 × 144	All	98.5	97.4	98.5	99.1	98.4
128 × 96	All	97.7	96.5	98.3	98.6	97.8
96 × 72	All	93.1	93.7	97.1	95.9	94.1
384 × 288	$x, \dot{x}, y, \dot{y}$	93.8	93.9	95.5	96.1	94.8

Table 3 Signer-independent isolated sign recognition with manual features in controlled environments. The *n*-best rate indicates the percentage of results for which the correct sign was among the *n* signs deemed most similar to the input sign by the classifier. For *n* = 1 this corresponds to what is commonly specified as the recognition rate

Training signer(s)	Test signer	Vocabulary size	*n*-Best rate		
			1 (%)	5 (%)	10 (%)
Michael	Sanchu	205	36.0	58.0	64.9
Paula, Sanchu	Michael	218	30.5	53.6	63.2
Ben, Paula, Sanchu	Michael	224	44.5	69.3	77.1
Ben, Michael, Paula	Sanchu	221	54.2	79.4	84.5
Ben, Michael, Sanchu	Paula	212	37.0	63.6	72.8
Michael, Sanchu	Ben	206	48.1	70.0	77.4

three signs were recognized just from non-manual features, a result that emphasizes the importance of facial expressions for sign language recognition.

6.2.1.2 Signer-independent recognition Table 3 shows the results for signer-independent recognition. Since the signers used different signs for some words, the vocabulary has been chosen as the intersection of the test signs with the union of all training signs. No selection has been performed otherwise, and no minimal pairs have been removed. As expected, performance drops significantly. This is caused by strong interpersonal variance in signing, as visualized in Fig. 47 by hand motion traces for identical signs done by different signers. Recognition rates are also affected by the exact constellation of training/test signers, and do not necessarily increase with the number of training signers.

Signer-independent performance in uncontrolled environments is difficult to measure, since it depends on multiple parameters (signer, vocabulary, background, lighting, camera). Furthermore, noise and outliers are inevitably introduced in the features when operating in real world settings. Despite the large interpersonal variance, signer-independent operation is feasible for small vocabularies, as can be seen in Table 4. Each test signer was recorded in a different real-life environment, and the selection of signs is representative of the complete vocabulary (it contains one-handed and two-handed signs, both with and without overlap).

6.2.2 Classification of continuous sign language

The continuous sign language recognition experiments were conducted on the DGS-Corpus described above. Table 5 present results for vocabulary sizes of 52, 97, and 152 signs, each with different employed language model.

Table 4 Signer-independent recognition rates in real-life environments

Vocabulary size	Test signer						
	Christian (%)	Claudia (%)	Holger (%)	Jörg (%)	Markus (%)	Ulrich (%)	∅ (%)
6	96.7	83.3	96.7	100	100	93.3	95.0
18	90.0	70.0	90.0	93.3	96.7	86.7	87.8

Analysing the results, it can be stated that the proposed system is able to recognize continuous sign language. Depending on the vocabulary size, sign accuracy ranges between 89.2 and 94.0% without language modeling (zerogram). In all cases, the utilization of uni- and bigram models can slightly improve recognition performance. Interestingly, increasing the vocabulary size by a factor of three does not worsen sign accuracy significantly.

Looking closer at the results, it is obvious that the system discriminates most of the minimal pairs, where the location, movement and orientation of the dominant hand are very similar. Another important aspect is the fact that the unseen sign successions in the test set are recognised in a good manner. Furthermore, the achieved recognition performance indicates that the system is able to handle the free order of signs within a sentence. Therefore, the system can be used for all aspects of sign language.

6.3 Recognition using subunit models

Performance evaluation of signer-dependent sign language recognition based on subunit models was carried out on the DGS-Corpus. Throughout all experiments only the manual features were employed. As already mentioned, recognition of continuous sign language requires a prohibitive amount of training material if signed sentences are used. Therefore the automated transcription of signs to subunits described in [2] was used here, which is expected to reduce the effort required for vocabulary extension to a minimum. Transcription was performed for 52 isolated signs, which resulted in 184 (pos), 187 (size) and 187 (dist) subunits respectively. The subsequent training is solely based on 5 repetitions of these isolated signs used for transcription.

Table 5 Signer-dependent recognition of continuous sign language using different language models (according to Sect. 4.3)

Vocabulary size	Language modeling		
	Zerogram ($m = 0$)	Unigram ($m = 1$)	Bigram ($m = 2$)
52	94.0%	94.7%	95.4%
97	91.8%	92.0%	93.2%
152	89.2%	89.6%	91.1%

Table 6 Signer-dependent recognition based on subunit models

Vocabulary size	Recognition rate (%)
52 (previously trained)	93.1
100 (previously not trained)	90.4

The obtained recognition results for isolated and continuous sign language are reported below.

6.3.1 Classification of isolated signs

In order to evaluate the automatic transcription of signs to subunits two different experiments were conducted (Table 6). In the first experiment recognition performance was tested for the 52 isolated signs used for transcription. It should be noted that the test set consists of the remaining five repetitions and thus differs from the training set, i.e., HMMs for subunits are trained with data of repetitions of the sign different from these used during this test. The recognition rate is 93.1%.

The second experiment simulates a later extension of the existing vocabulary by 100 additional signs. No data of any repetition was used to train HMMs for subunits. As these are signs which were newly included into the vocabulary, their transcriptions were determined after the completion of the training. The performed classification test for these signs synthesized from the identified subunits yields 90.4% correct recognition. This result shows that the automatic segmentation of signs produced reasonable subunits.

6.3.2 Classification of continuous sign language

The experiments carried out for continuous sign language recognition are based on a test set of 100 sentences. These sentences were chosen from the DGS-Corpus in such a way that they only comprise the 52 signs used for transcription. Each sentence ranges from two to nine signs in length. After eliminating coarticulation effects and considering a bigram model according to the statistical probability of sign order, a recognition rate of 87.7% was achieved. Without language model, accuracy drops to 80.3%. This finding is very essential since it solves the problem of vocabulary extension without additional training.

6.4 Experimental results for signer adaptation

The performance evaluation for signer adaptation is limited to supervised adaptation only. The undertaken experiments were conducted on the base vocabulary of the BSL-Corpus.

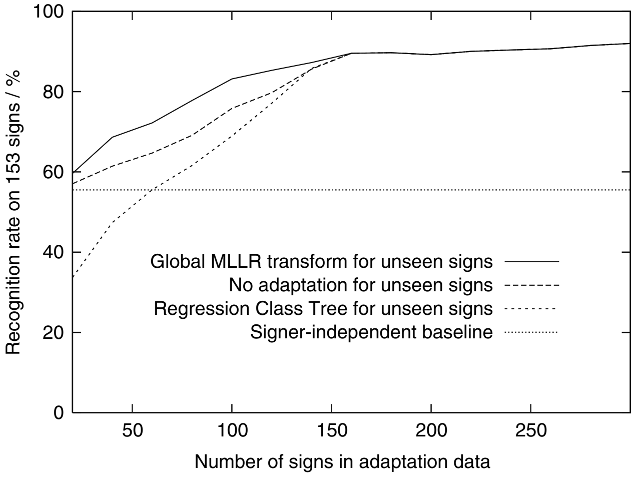


Fig. 51 Handling of unseen signs, supervised MLLR adaptation

Three signers were used for training the signer-independent model, and one signer served for testing. Four repetitions were used for static adaptation with different amounts of adaptation data while one repetition was reserved for testing. All results given are average values from the four possible combinations of signers (leaving-one-out tests).

Only the Gaussian means were updated by MLLR and MAP, variances and mixture weights remain unchanged as the mean covers most of the variability between the signers [23]. The results below are derived using Gaussian single densities, experiments with Gaussian mixtures show the same behavior due to the small training population. Explicit one-hand transformations were used in all MLLR experiments.

Figure 51 illustrates the effect of the proposed methods for handling mixture components from unseen signs. Seen components were adapted with the most special transform

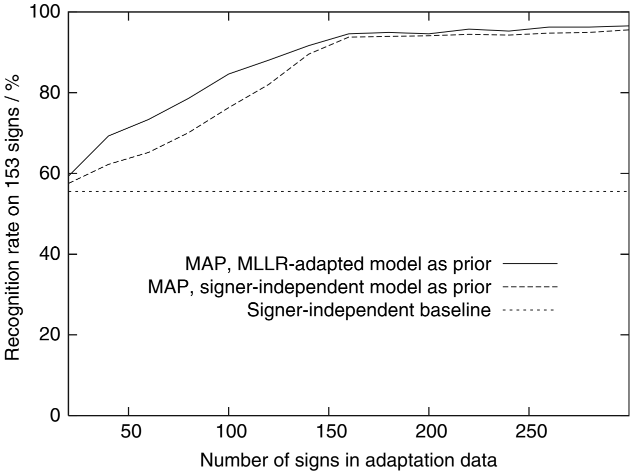


Fig. 52 Supervised MAP: Signer-independent versus MLLR-adapted model as prior

Table 7 Results for supervised adaptation

Method	Recognition rate on 153 signs, number of adaptation utterances		
	80	160	320
Signer-dependent	97.9%	97.9%	97.9%
Signer-independent	55.5%	55.5%	55.5%
MAP	70.1%	93.8%	95.9%
MLLR	77.8%	89.5%	91.7%
MLLR→MAP	78.6%	94.6%	96.9%

from the regression class tree in all three experiments. As described, transforming the unseen components with a global transformation outperforms the conventional approach and is superior to ignoring the unseen components during adaptation. Therefore, the MLLR approach is suited for rapid signer adaptation using only a small amount of adaptation data.

Combining the modified MLLR approach with standard MAP, as shown in Fig. 52, results in the same effect which has been observed in the field of speech recognition: the rapid adaptation using MLLR is preserved, while its saturation is compensated by MAP.

Table 7 summarizes the adaptation experiments, showing the recognition performance of adapted models using the different methods. MLLR followed by MAP yields the best models, regardless of the number of adaptation sequences. Using class-based MLLR, rapid adaptation to an unknown signer is possible without covering the entire vocabulary during adaptation, as described in [28].

7 Conclusions

This paper has described a comprehensive approach to robust visual sign language recognition which reflects recent developments in this field. The proposed recognition system aims to signer-independent operation and utilizes a single video camera for data acquisition to ensure user-friendliness. In order to cover all aspects of sign languages, sophisticated algorithms were developed that robustly extract manual and facial features, also in uncontrolled environments. The classification stage is designed for recognition of isolated signs as well as of continuous sign language. For statistical modeling of reference models, a single sign can be represented either as a whole or as a composition of smaller subunits—similar to phonemes in spoken languages. In order to overcome the problem of high interpersonal variance, dedicated adaptation methods known from speech recognition were implemented and modified to consider the specifics of sign languages.

Remarkable recognition performance has been achieved for signer-dependent classification and medium sized vocabularies. Furthermore, the presented recognition system is suitable for signer-independent real world applications where small vocabularies suffice, as, e.g., for controlling interactive devices. Breaking down signs into smaller subunits allows the extension of an existing vocabulary without the need of large amounts of training data. This constitutes a key feature in the development of sign language recognition systems supporting large vocabularies. Methods for signer adaptation yields significant performance improvements. While the modified maximum likelihood linear regression approach serves for rapid adaptation to unknown signers, the combined maximum a posteriori estimation results in high accuracy for larger sets of adaptation data.

Acknowledgments The presented work was partially supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) and the European Commission, Directorate General Information Society, IST–2000–27512 WISDOM. The videos containing British Sign Language were kindly provided by the British Deaf Association.

References

1. Bahl, L., Jelinek, F., Mercer, R.: A maximum likelihood approach to continuous speech recognition. IEEE Trans. Pattern Anal. Mach. Intell. **5**(2), 179–190 (1983)
2. Bauer, B.: Erkennung kontinuierlicher Gebärdensprache mit Untereinheiten-Modellen. Shaker Verlag, Aachen (2003)
3. Becker, C.: Zur Struktur der deutschen Gebärdensprache. WVT Wissenschaftlicher Verlag, Trier (Germany) (1997)
4. Canzler, U.: Nicht-intrusive Mimikanalyse. Dissertation Chair of Technical Computer Science, RWTH, Aachen (2005)
5. Canzler, U., Dziurzyk, T.: Extraction of non manual features for videobased sign language recognition. In: Proceedings of the IAPR Workshop on Machine Vision Applications, pp. 318–321. Nara, Japan (2002)
6. Canzler, U., Ersayar, T.: Manual and facial features combination for videobased sign language recognition. In: Proceedings of the 7th International Student Conference on Electrical Engineering. Prague (2003)
7. Canzler, U., Kraiss, K.-F.: Person-adaptive facial feature analysis for an advanced wheelchair user-interface. In: Conference on Mechatronics and Robotics, vol. Part III, pp. 871–876. Sascha Eysoldt Verlag (2004)
8. Canzler, U., Wegener, B.: Person-adaptive facial feature analysis. In: Proceedings of the 8th International Student Conference on Electrical Engineering. Prague (2004)
9. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)
10. Derpanis, K.G.: A review of vision-based hand gestures. Technical Report, Department of Computer Science, York University (2004)
11. Dick, T., Zieren, J., Kraiss, K.-F.: Visual hand posture recognition in monocular image sequences. In: Pattern Recognition, 28th DAGM Symposium Berlin, Lecture Notes in Computer Science. Springer, Berlin (2006)
12. Fang, G., Gao, W., Chen, X., Wang, C., Ma, J. Signer-independent continuous sign language recognition based on SRN/HMM.

In: Revised Papers from the International Gesture Workshop on Gestures and Sign Languages in Human–Computer Interaction, pp. 76–85. Springer, Heidelberg (2002)

13. Gales, M., Woodland, P.: Mean and variance adaptation within the MLLR framework. Comput. Speech Lang. **10**, 249–264 (1996)
14. Hermansky, H., Timberwala, S., Pavel, M.: Towards ASR on partially corrupted speech. In: Proceedings of the 4th International Conference on Spoken Language Processing, vol. 1, pp. 462–465. Philadelphia, PA (1996)
15. Holden, E.J., Owens, R.A.: Visual sign language recognition. In: Proceedings of the 10th International Workshop on Theoretical Foundations of Computer Vision, pp. 270–288. Springer, Heidelberg (2001)
16. Huang, X., Ariki, Y., Jack, M.: Hidden Markov Models for Speech Recognition. Edinburgh University Press, Edinburgh (1990)
17. Illingworth, J., Kittler, J.: A survey of the Hough transform. Computer Vision, Graphics, and Image Processing **44**(1), 87–116 (1988)
18. Imai, A., Shimada, N., Shirai, Y.: 3-D hand posture recognition by training contour variation. In: Proceedings of the International Conference on Automatic Face and Gesture Recognition (2004)
19. Jelinek, F.: Statistical Methods for Speech Recognition. MIT, Cambridge (1998). ISBN 0-262-10066-5
20. Jones, M., Rehg, J.: Statistical color models with application to skin detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab (1998)
21. Kraiss, K.-F. (ed): Advanced man–machine interaction. Springer, Heidelberg (2006). ISBN 3-540-30618-8
22. Lee, C.-H., Lin, C.-H., Juang, B.-H.: A study on speaker adaptation of the parameters of continuous density hidden Markov models. IEEE Trans. Acoust. Speech Signal Process. **39**(4), 806–814 (1991)
23. Leggetter, C.J.: Improved acoustic modelling for HMMs using linear transformations. Ph.D. Thesis, Cambridge University (1995)
24. Liang, R.H., Ouhyoung, M.: A real-time continuous gesture interface for Taiwanese sign language. In: Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology. Banff, Alberta, Canada, 14–17 October 1997
25. Liddell, S.K., Johnson, R.E.: American sign language: the phonological base. Sign Lang. Stud. **18**(64), 195–277 (1989)
26. Lievin, M., Luthon, F.: Nonlinear color space and spatiotemporal MRF for hierarchical segmentation of face features in video. IEEE Trans. Image Process. **13**, 63–71 (2004)
27. Murakami, K., Taguchi, H.: Gesture recognition using recurrent neural networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 237–242. ACM, New York (1991)
28. Ong, S.C.W., Ranganath, S.: Deciphering gestures with layered meanings and signer adaptation. In: Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (2004)
29. Ong, S.C.W., Ranganath, S.: Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 873–891 (2005)
30. Parashar, A.S.: Representation and interpretation of manual and non-manual information for automated American sign language recognition. Ph.D. Thesis, Department of Computer Science and Engineering, College of Engineering, University of South Florida (2003)
31. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
32. Rabiner, L.R., Juang, B.-H.: An introduction to hidden Markov models. IEEE Acoust. Speech Signal Process. Soc. Mag. **3**(1), 4–16 (1986)
33. Rabiner, L.R., Juang, B.-H.: Fundamentals of Speech Recognition. Prentice-Hall, Upper Saddle River, ISBN 0-13-015157-2 (1993)
34. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. International Thomson Publishing (1998). ISBN 0-534-95393-X
35. Starner, T., Weaver, J., Pentland, A.: Real-time American sign language recognition using desk and wearable computer based video. IEEE Trans. Pattern Anal. Mach. Intell. **20**(12), 1371–1375 (1998)
36. Stokoe, W.: Sign language structure: an outline of the visual communication systems of the american deaf. (Studies in Linguistics. Occasional paper, University of Buffalo (1960)
37. Sturman, D.J.: Whole-hand input. Ph.D. Thesis, School of Architecture and Planning, Massachusetts Institute of Technology (1992)
38. Sutton, V.: http://www.signwriting.org/ (2003)
39. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical Report CS-91-132, CMU, 1991
40. Vamplew, P., Adams, A.: Recognition of Sign Language Gestures Using Neural Networks. In: European Conference on Disabilities, Virtual Reality and Associated Technologies (1996)
41. Vittrup, M., Sørensen, M.K.D, McCane, B.: Pose Estimation by Applied Numerical Techniques. Image and Vision Computing, New Zealand (2002)
42. Vogler, C., Metaxas, D.: Parallel hidden Markov models for American sign language recognition. In: Proceedings of the International Conference on Computer Vision (1999)
43. Vogler, C., Metaxas, D.: Toward scalability in ASL recognition: breaking down signs into phonemes. In: Gesture-Based Communication in Human–Computer Interaction, International Gesture Workshop, GW'99, Lecture Notes in Computer Science, pp. 211–224. Springer, Berlin (1999)
44. von Agris, U., Schneider, D., Zieren, J., Kraiss, K.-F.: Rapid signer adaptation for isolated sign language recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop. New York, USA (2006)
45. Welch, G., Bishop, G.: An introduction to the Kalman Filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill (2004)
46. Yang, M., Ahuja, N., Tabb, M.: Extraction of 2D motion trajectories and its application to hand gesture recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1061–1074 (2002)
47. Zieren, J., Kraiss, K.-F.: Robust person-independent visual sign language recognition. In: Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis, Lecture Notes in Computer Science (2005)

48 Zieren, J.: Visuelle Erkennung von Handposituren für einen interaktiven Gebärdensprachtutor. Dissertation, Chair of Technical Computer Science, RWTH Aachen (2007)