Chapter 2
Hierarchically Structured Non-Intrusive Sign Language Recognition

***

Jorg Zieren and Karl-Friedrich Kraiss
Chair of Technical Computer Science
RWTH Aachen University
fzieren,kraissg@techinfo.rwth-aachen.de
www.techinfo.rwth-aachen.de

Abstract

This work presents a hierarchically structured approach at the nonintrusive recognition of sign language from a monocular frontal view. Robustness is achieved through sophisticated localization and tracking methods, including a combined EM/CAMSHIFT overlap resolution procedure and the parallel pursuit of multiple hypotheses about hands position and movement. This allows handling of ambiguities and automatically corrects tracking errors. A biomechanical skeleton model and dynamic motion prediction using Kalman filters represents high level knowledge. Classification is performed by Hidden Markov Models. 152 signs from German sign language were recognized with an accuracy of 97.6%.

1. Introduction

Manual gestures are an important information carrier in everyday communication. Considerable potential lies in the automatic recognition of gestures, especially for human-computer interaction. As opposed to the keyboard, gestures are natural, intuitive, and do not require special skills. Sign language recognition is a particularly challenging field in this research area. Its goal is to do for deaf people what speech recognition has done for hearing people: Offer the most natural way of controlling electronic devices.

In sign language, information is communicated primarily through hands and face. This work utilizes manual parameters for the recognition of 152 signs from German sign language. Instead of using data gloves, which in most scenarios is not an acceptable solution, manual parameters are extracted from images acquired by a single video camera positioned in front of the signer. Skin color and motion form the basic low-level image cues. In order to ensure a natural, i. e. non-intrusive interaction, no other devices, such as markers or additional cameras, are employed.

Existing non-intrusive systems only support considerably smaller vocabularies of about 40 signs [12]. Due to the difficulty of accurately localizing the signer's hands when they are overlapping with each other and/or with the face, which occurs frequently in sign language, ambiguities can easily arise. Sophisticated overlap resolution procedure and the parallel pursuit of multiple hypotheses regarding hands position and movement, are applied to compute manual features even in such problematic scenes. *A. priori* knowledge is incorporated through a biomechanical skeleton model and dynamic Kalman filter predictions.

The extracted features are classified using Hidden Markov Models (HMMs) to compensate for variations in speed and allow limited variations in amplitude. On the chosen data set the developed system achieves a recognition rate of 97.6% at a resolution of 384 *X* 288 pixels. This performance has been measured for a person dependent recognition task in a controlled environment. However, the system's basic concepts are not geared towards this scenario. Their suitability for "real life", possibly mobile environments, is an important design feature.

2. Sign Vocabulary

The system's vocabulary consists of 152 signs. Each has been recorded ten times with a resolution of 384 *X* 288 pixelS and 25 frames per second. Figure 1 shows an example sign and the recording conditions, which were identical for all signs. Since this work focuses on person-dependent classification, only one signer has been recorded. Extending the system to person-independent classification would not affect the tracking stage, but inter-personal variance would require special measures in the classification stage if comparable recognition rates were to be achieved.

Figure 1. Example sign "computer" from the test/training data set. Signer, background, illumination, and clothing were identical in all recordings.

3. Tracking

The complexity of the object tracking task suggests a hierarchical division in two stages (see Figure 2). First, a low level processing stage detects a set of target candidates using skin color as an image cue. This set may include skin colored distracters. Hands and face are then found in this set by the subsequent high level processing stage. To this end, multiple hypotheses are evaluated per frame and over time.

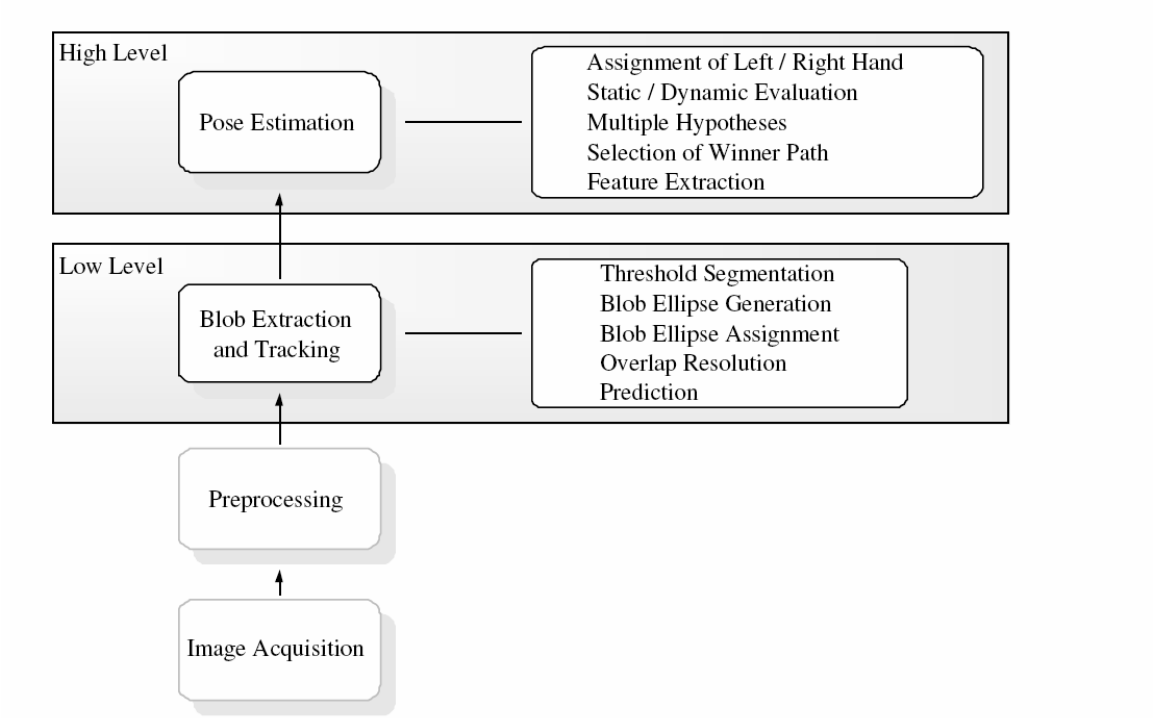


Figure 2. Functional overview.

3.1 Target Candidate Detection

Based on generic skin and non-skin color histograms presented in [7], a skin color probability is computed for every pixel. After smoothing the result with a Gaussian kernel and application of a threshold segmentation [13], contiguous regions (blobs) are extracted.

The computationally demanding high level stage necessitates efficient data structures for the representation of each blob. Therefore, a blob's boundary (which typically consists of several hundred pixels) is not processed directly, but approximated by an elliptical representation called "blob ellipse." Aiming for a tradeoff between accuracy in terms of the signal to noise ration and processing speed, a blob ellipse is described by its center coordinates x and y, radii r_a and r_b, and orientation of the principal axis.

It is obvious that a threshold segmentation cannot separate two or more overlapping skin colored objects (e. g. hand and face). To extract meaningful features, however, a separation of the overlapping objects is required. Therefore, a distinction is introduced between the set of "raw" blob ellipses extracted in frame t, called $B_{raw;t}$, and a corresponding set of "overlap resolved" blob ellipses B_t. Only B_t will later be forwarded to the high level stage. This is illustrated in

Figure 3. In the input image I_{t-1}, no overlap is present. Therefore, $B_{raw;t-1} = B_{t-1}$. In I_t, the right hand is overlapping with the face. In $B_{raw;t}$, the two corresponding ellipses have therefore merged into one. The low level stage resolves this overlap and computes two overlapping ellipses. This process is described in the following section.

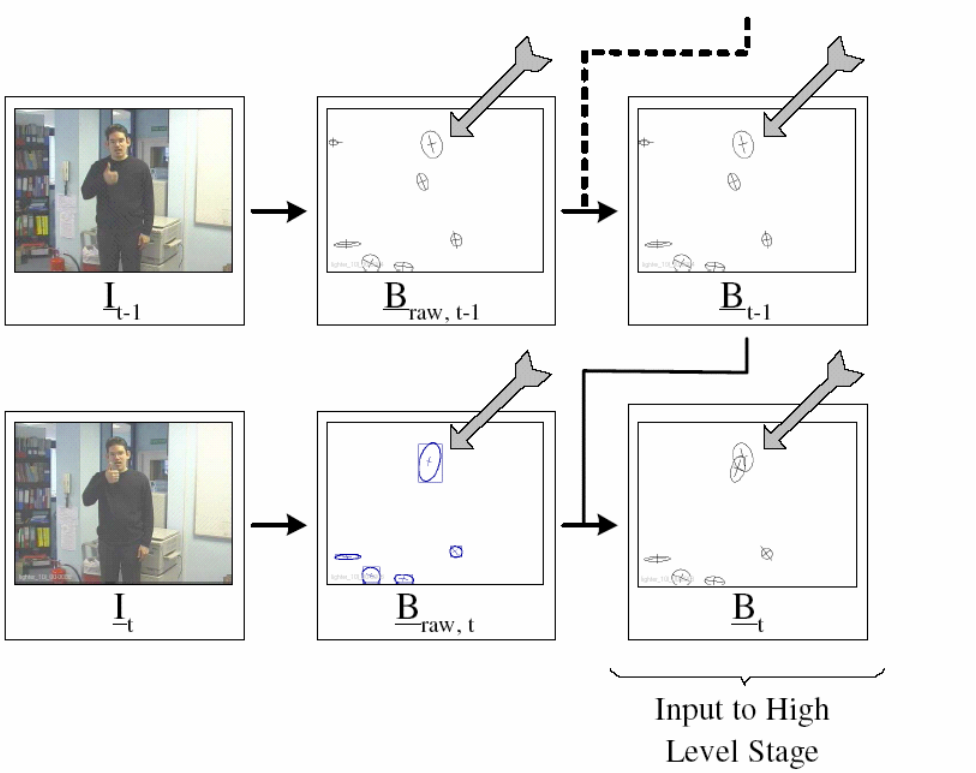


Figure 3. Processing of blob ellipses by the low level stage. Grey arrows indicate overlap resolution.

3.1.1 Overlap Detection and Resolution

For each blob ellipse in $B_{raw;t}$, a number of n an element of N_0 corresponding blob ellipses in B_{t-1} are found by computing and evaluating predictions for both shape and position. Depending on n, several cases can be distinguished as shown in Table 1.

Table 1.
Different cases of blob ellipse correspondence.

n = 0	new object has entered the image
n = 1	regular tracking
n ≥ 2	*n* objects have started to overlap

For n ≥ 2, either the EM or the CAMSHIFT algorithm is used to resolve the overlap and approximate features for all overlapping objects. This is described below.

3.1.2 Overlap Resolution Using the EM Algorithm

The Expectation Maximization (EM) algorithm is an iterative method for approximating a given probability distribution by a superposition of a fixed number of two-dimensional Gaussian distributions [2]. The latter corresponds well with the concept of blob ellipses, which allows an easy integration of the EM algorithm in the processing chain.

Figure 4 shows a typical scenario that can be treated with the EM algorithm. The overlap here is only partial, i. e. none of the three overlapping objects is completely enclosed in another object (in a 2D sense). Since the threshold segmentation yields a binary mask, but the EM algorithm computes a superposition of Gaussians, a morphological distance transformation (described in [6]) is used to create a pseudo-multivariate distribution. This requires that non-skin colored pixels (holes) enclosed by the overlapping objects are first removed, i. e. set to 1 in the binary mask.

Figure 4. Preparation of the skin color mask for the EM algorithm. (a) Original image, (b) Threshold segmented skin color probability, (c) Skin color mask with holes removed, (d) Distance transformed skin color mask.

For the EM algorithm to accurately resolve an overlap of multiple blob ellipses, t is initialized with the shape and position parameters computed for these ellipses in the previous frame. The original algorithm has been modified so that several parameters either remain constant or change only in a well-defined interval. This increases the stability of the approximation process. Figure 5 shows the approximation status at different iterations.

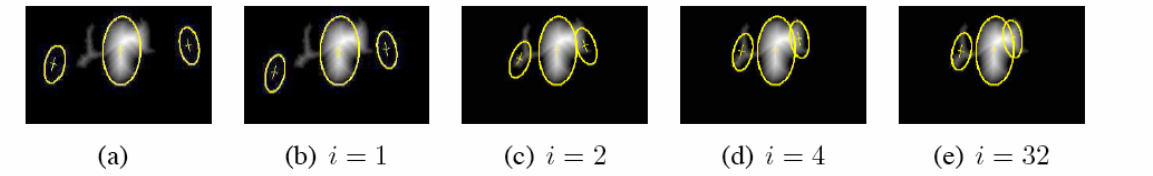


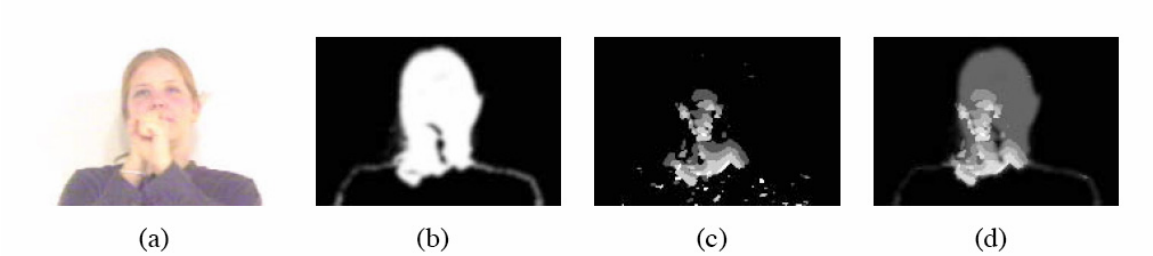
Figure 5. Application of the EM algorithm. (a) Initialization, (b – e) after *i* iterations.

3.1.3 Overlap Resolution Using the CAMSHIFT Algorithm

If one object is completely enclosed in another, the EM algorithm is unsuitable for overlap resolution because its input matrix (Figure 4d) would not provide any information about the inner object. However, motion can be used as an additional image cue in this case. The detection of motion is based on computing, for every pixel, the difference in color between successive frames, and subsequent application of a fixed threshold to yield a binary "motion mask." Using a sliding average with linearly decreasing weights, a so called Motion History Image (MHI), $I_{motion}(x; y)$, is created as described in [3]. This is then combined with the skin color probability distribution $I_{skin}(x; y)$ according to the following equations. Figure 6 shows a visualization of this process.

$$I_{skin}(x; y) = p_{skin}(x; y) \quad (1)$$

$$I_{camshift}(x; y) = w\, I_{skin}(x; y) + (1 ¡ w)\, I_{motion}(x; y) \quad (2)$$



(a) (b) (c) (d)

Figure 6. Computation of the CAMSHIFT input image. (a) Original image, (b) Skin color probability image (Gauss filtered), (c) Motion History Image, (d) Combined image according to equation 2.

On the resulting image $I_{camshift}$, a CAMSHIFT tracker is applied for each hand. The respective search windows are initialized with the most recent position and shape values of the overlapping blob ellipses.

Shape and orientation remain constant while this method is used for overlap resolution. The weight w ($0 \cdot w \cdot 1$) allows adjustment of the degree to which motion is considered by the CAMSHIFT algorithm. Figure 7 shows an example application to a face-hand-hand overlap.

(a) t=13 (b) t=18 (c) t=23

Figure 7. Resolution of a face-hand-hand overlap by application of the CAMSHIFT algorithm.

3.2 Multiple Hypothesis Tracking

From the set of detected target candidates, the actual body configuration that gave rise to this observation has to be deduced. Since every observation allows more than one interpretation, multiple hypotheses can be formulated for every video frame. Enumerating these hypotheses and plotting them over time results in a diagram as shown in Figure 8. In this hypothesis space, there are $N(t)$ hypotheses for frame t, thus the total number of all possible paths (i.e. tracking results) P equals

$$P = \prod_{t} N(t) \qquad (3)$$

High level knowledge about the signing process allow computing for each hypothesis a probability $p_{stat;t}(i)$ which is independent from the previous and the next hypothesis (static), and a probability $p_{dyn;t}(i; j)$ which depends only on the transition between two hypotheses, but not on the hypotheses themselves (dynamic). Searching for the path with highest total probability is done with the Viterbi algorithm [9].

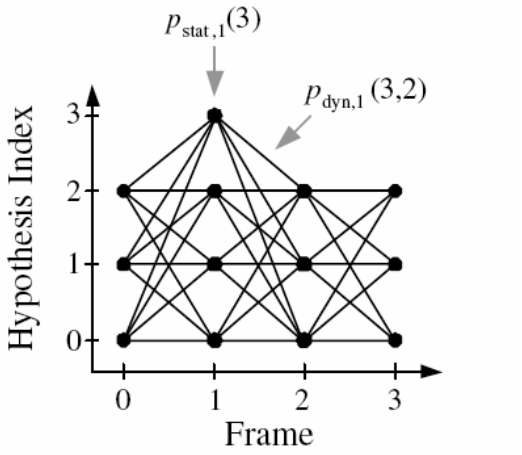


Figure 8. Hypothesis space with static and dynamic probabilities.

3.2.1 Computation of Static and Dynamic Probabilities

The chosen approach allows exploitation of any number of image cues and high level knowledge for the computation of the static and dynamic probabilities. In practice, the selection will depend on the actual recognition/tracking task. For the presented system, a body model is computed that approximates arm length and flexion of joints. From a manual segmentation of the input clips, a hand position histogram has been created that represents knowledge of where the hands are typically found. Together with information about the signer's handedness, this allows to evaluate the hypothesized configuration's likeliness both physiologically and "linguistically," resulting in p_{stat}. p_{dyn} is obtained from Kalman filter predictions for each blob ellipse's position, shape, and orientation.

3.3 Feature Vector Composition

The feature vector composed for every frame contains the center coordinates, orientation, area, and ratio of radii of each hand's elliptical approximation. Furthermore, compactness and eccentricity (as defined in [10]) are computed from the object's border found by the threshold segmentation. For the feature vector to be independent from the signer's exact position in the image and from the camera's resolution, position and area are specified relative to the face position and face width. Derivatives for all of these values are also added to the feature vector.

4. Evaluation

Since the tracking stage is the most complex component in this work, not only the recognition rate, but also the tracker's hit rate were evaluated. A manual segmentation of all input clips has been performed which allows to define three categories that classify a tracking result based on the center coordinates as shown in Table 2 and Figure 9.

Table 2. Categorization of tracking results.

Center coordinates (*x, y*) within...	Category
...border of the target object	hit on object
...elliptical region around target center	hit near center
...neither of the above	miss

Figure 9. Definition of tracker hit and miss.

Experimental results are shown in Table 3. The vocabulary has been divided into five categories that clearly show overlap to be the tracker's main problem.

Table 3. Quantitative evaluation of tracking accuracy (H=hand, F=face).

Sign Category	One Handed	Two Handed	No Overlap	H-F Overlap	H-H Overlap	Total
Hit rate:	98.4%	95.4%	99.0%	96.8%	93.7%	97.1%

On the complete vocabulary of 152 signs in German Sign Language, a recognition rate of 97.6% was achieved using an HMM based classification stage. This constitutes an increase compared to other recognition systems (intrusive and nonintrusive), such as [4], [5], [11], and [12]. Only for considerably smaller vocabularies (approx. 40 signs) have higher rates for non-intrusive recognition been published. This may be due to the fact that the multiple hypotheses approach considers for every frame nearly all available information, including past and future frames, before a decision is made, and can retrospectively correct tracking errors as soon as they become apparent.

5. Outlook

Several improvements and extensions are conceivable to either increase recognition performance or open up new application scenarios. A user adaptive skin color model would reduce the number of distracters by narrowing down the target color range and thereby increasing both reliability and processing speed of the tracking stage.

Significantly increasing the vocabulary size would require the extraction of further shape and/or texture features, with the ultimate goal of reconstructing a 3D hand model from the 2D image data.

Recognition of continuous signing is an obvious but complex extension of the system. Translation systems to speech, text, or another sign language, require the automatic detection of start and end points of individual signs, as well as the handling of co-articulation effects that can have strong influence on the extracted features.

Integration of mimic, i. e. facial features, is currently in progress. Facial expressions are vital for sign language recognition since many signs are identical in their manual features. A further increase in recognition rates can be expected from this extension.

Acknowledgements

This work was carried out at the Chair of Technical Computer Science, RWTH Aachen University, based on a dissertation by Suat Akyol [1]. Numerous other researchers and students have contributed code to the developed software [8]. The project is funded by the European Commission Directorate – General Information Society Technologies (IST) Programme (2001–2003)

References

[1] S. Akyol. *Nicht-intrusive Erkennung isolierter Gesten und Gebaerden (Non-Intrusive Recognition of Isolated Gestures and Signs)*. 2003. Dissertation, Chair of Technical Computer Science, RWTH Aachen University.

[2] J. A. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussion Mixture and Hidden Markov Models. Technical Report TR-97-021, International Computer Science Institute, U.C. Berkeley, April 1998.

[3] J.-W. Davis and A.-F. Bobick. The Representation and Recognition of Action Using Temporal Templates. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 928–934, San Juan, Puerto Rico, 1997.

[4] K. Grobel. *Videobasierte Geb¨ardenspracherkennung mit Hidden-Markov- Modellen (Video-Based Sign Language Recognition Using Hidden Markov Models)*. Fortschritts-Berichte VDI 10/592. VDI Verlag, D¨usseldorf, 1999. Dissertation, Chair of Technical Computer Science, RWTH Aachen University.

[5] H. Hienz. *Erkennung kontinuierlicher Geb¨ardensprache mit Ganzwortmodellen (Recognition of Continuous Sign Language Using Whole Word Models)*. Shaker Verlag, Aachen, 2000. Dissertation, Chair of Technical Computer Science, RWTH Aachen University.

[6] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1989.

[7] M. J. Jones and J. M. Rehg. Statistical Color Models with Application to Skin Detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab, December 1998.

[8] LTI-Lib: A C++ library for image processing and computer vision. http://ltilib.sf.net, 2003.

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

[10] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing Company, 1999.

[11] T. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk andWearable Computer Based Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[12] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2D Motion Trajectories and its Application to Hand Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074, 2002.

[13] J. Zieren, N. Unger, and S. Akyol. Hands Tracking from Frontal View for Vision-Based Gesture Recognition. In L. van Gool, J. Hartmanis, and J. van Leeuwen, editors, *Lecture Notes in Computer Science LNCS 2449*, Z¨urich, Switzerland, 2002.